



# Détection de structures fines par traitement d'images et apprentissage statistique : application au contrôle non destructif

Vincent Morard

## ► To cite this version:

Vincent Morard. Détection de structures fines par traitement d'images et apprentissage statistique : application au contrôle non destructif. Mathématiques générales [math.GM]. Ecole Nationale Supérieure des Mines de Paris, 2012. Français. NNT : 2012ENMP0101 . pastel-00932727

**HAL Id: pastel-00932727**

**<https://pastel.archives-ouvertes.fr/pastel-00932727>**

Submitted on 17 Jan 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale n°432 :  
Sciences des Métiers de l'Ingénieur

**Doctorat ParisTech**

**T H È S E**

pour obtenir le grade de docteur délivré par

**l'École nationale supérieure des mines de Paris**

**Spécialité « Morphologie Mathématique »**

*présentée et soutenue publiquement par*

**Vincent MORARD**

22 octobre 2012

**Détection de structures fines par traitement d'images et  
apprentissage statistique : application au contrôle non  
destructif**

Thèse confidentielle jusqu'en octobre 2029

Directeur de thèse : **Etienne DECENCIÈRE**  
Co-encadrant de thèse : **Petr DOKLÁDAL**

**Jury**

**M. Etienne DECENCIÈRE**, Maître de recherche, CMM-MS, Mines ParisTech  
**M. Petr DOKLÁDAL**, Docteur, CMM-MS, Mines ParisTech  
**M. Marc VAN DROOGENBROECK**, Professeur, INTEL SIG, Université de Liège  
**M. Dominique JEULIN**, Professeur, CMM-MS, Mines ParisTech  
**M. Claude LEONETTI**, Président-directeur général, T.P.S.H.  
**M. Philippe SALEMBIER**, Professeur, UPC, Université polytechnique de Catalogne  
**M. Hugues TALBOT**, Docteur, A3SI, Université Paris-Est

Examineur  
Examineur  
Rapporteur  
Examineur  
Examineur  
Rapporteur  
Examineur

**MINES ParisTech**  
**Centre de Morphologie Mathématique, Mathématiques et Systèmes**  
35, rue Saint-Honoré, 77305 Fontainebleau

**T  
H  
È  
S  
E**



**Détection de structures fines par traitement d'images et apprentissage statistique : application au contrôle non destructif**

**Résumé :** Dans cette thèse, nous présentons de nouvelles méthodes de traitement d'images pour extraire ou rehausser les éléments fins d'une image. Pour ces opérateurs, issus de la morphologie mathématique, l'accent a été mis principalement sur la précision de détection et sur le temps de calcul, qui doivent être optimisés pour pouvoir répondre aux contraintes de temps imposées par différentes applications industrielles. La première partie de ce mémoire présente ces méthodes, organisées en fonction de la tortuosité des objets à détecter. Nous commençons par proposer un algorithme rapide pour le calcul des ouvertures 1-D afin d'extraire des structures rectilignes des images. Puis, nous étudions une nouvelle classe d'opérateurs rapides avec les ouvertures parcimonieuses par chemins, permettant d'analyser des structures ayant une tortuosité modérée. Enfin, nous proposons de nouveaux éléments structurants adaptatifs et des filtres connexes construits avec des attributs géodésiques et géométriques pour extraire des structures filiformes ayant une tortuosité quelconque.

Dans un second temps, nous avons développé une méthode d'analyse statistique en introduisant une nouvelle pénalisation adaptative. L'objectif consiste à créer un modèle prédictif précis, qui minimise en même temps une fonction de coût, indépendante des données. Lorsque cette fonction de coût est liée au temps de calcul de chaque descripteur, il est alors possible de créer un modèle parcimonieux précis et qui minimise les temps de calcul. Cette méthode est une généralisation des régressions linéaires et logistiques Ridge, Forward stagewise, Lar, ou Lasso.

Les algorithmes développés dans cette thèse ont été utilisés pour trois applications industrielles, très différentes les unes des autres, mais toutes faisant intervenir une approche multidisciplinaire : le traitement d'images et l'analyse statistique. L'association de ces deux disciplines permet d'améliorer la généricité des stratégies proposées puisque les opérateurs de traitement d'images alliés à un apprentissage supervisé ou non supervisé, permettent d'adapter le traitement à chaque application.

**Mots clés :** Traitement d'images, morphologie mathématique, analyse statistique, caractérisation de formes, contrôles non destructifs, ouvertures parcimonieuses par chemins, region growing structuring elements (REGSE), amincissements par attributs géodésiques et géométriques, Adaptive coefficient shrinkage (AdaCOS).





**Thin structures detection by means of image processing and statistical learning :  
application to non-destructive testing**

**Abstract :** This PhD is dedicated to new image processing methods to extract or enhance thin objects from an image. These methods stem from mathematical morphology, and they mainly focus on the accuracy of the detection and on the computation time. This second constraint is imposed by the fact that we are dealing with high-throughput applications. The first part of this thesis presents these methods, organized according to the tortuosity of the objects to detect. We first propose a fast algorithm for the computation of 1-D openings, used to extract thin and straight structures in the images. Then, we study a new class of fast operators, parsimonious path openings, which can extract thin structures with moderate tortuosities. Finally, we propose new adaptive structuring elements and new thinnings with geodesic and geometric attributes to filter out the noise and to enhance thin structures of any tortuosity.

Besides, we have developed a machine learning method by introducing a new adaptive penalization. We aim at creating a predictive model that minimizes a cost function (independent of the data) while preserving a good accuracy. When this cost function is linked to the computation time of each feature, the resulting models will optimize the timings, while preserving a good accuracy. This method is a generalization of linear and logistic regressions with Ridge, Forward stagewise, Lar or Lasso penalization.

The algorithms developed in this thesis have been used for three industrial applications. While their objectives are very different, the framework is the same (non-destructive testing) and they all involve a multidisciplinary approach (images processing and statistical analysis). The combination of these two fields yields a higher flexibility in comparison with classical methods. Generic strategies are used, since image processing operators are associated to statistical learning (supervised or unsupervised) to make a specific treatment for each application.

**Keywords :** Image processing, mathematical morphology, statistical analysis, pattern recognition, non destructive testing, parsimonious path openings, region growing structuring elements (REGSE), geodesic and geometric attributes thinnings, Adaptive coefficient shrinkage (AdaCOS).

À ma famille,  
À Virginie.

---

## Remerciements

---

Cette thèse est le *fruit* de trois années de travail au centre de morphologie mathématique, laboratoire de l'école des Mines de Paris. Aussi, je tiens à exprimer, par ces quelques lignes, ma sincère reconnaissance envers tous ceux qui m'ont entouré, encadré et soutenu tout au long de cette thèse et qui ont contribué à rendre ce *fruit* le plus mature possible.

Je tiens à remercier Fernand Meyer, directeur du centre de morphologie mathématique, pour m'avoir accueilli dans son équipe, pour sa sagesse et son expérience qu'il sait transmettre à chacun.

Mes vifs remerciements viennent ensuite à Dominique Jeulin pour son savoir immense et pour m'avoir fait confiance tout au long de cette thèse et ce, dès le premier jour.

Je remercie Marc Van Droogenbroeck, professeur à l'université de Liège, et Philippe Salembier, professeur à l'université polytechnique de Catalogne, d'avoir accepté de participer à ce jury en tant que rapporteurs. Les remarques et commentaires ont contribué à l'amélioration de ce manuscrit.

Je tiens à remercier Hugues Talbot, chercheur à l'université Paris-Est, pour m'avoir fait l'honneur de présider le jury de ma soutenance et Claude Leonetti, président-directeur général de TPSH, pour avoir accepté de faire partie du jury.

J'ai souvent entendu que de bonnes relations doctorant-encadrants étaient la clef de voute dans la réussite d'une thèse. J'en ai maintenant la confirmation. Je n'aurai pas pu rêver de meilleurs encadrants, en les personnes d'Etienne Decencière et de Petr Dokládál, pour bâtir cette thèse. Ils ont su diriger et orienter mes recherches toujours dans la bonne direction, tout en me laissant une liberté d'action et une autonomie très bénéfique. Sans leurs précieux conseils, leurs clairs voyances, leurs critiques et leurs encouragements ce travail n'aurait pas été possible. Etienne, Petr, vous avez été de véritables mentors pour moi. Merci !

A présent, mes remerciements vont vers le personnel du CMM qui ont rendu ce séjour Bellifontain si agréable et si enrichissant. Je suis très reconnaissant envers Catherine Moysan, si prompte à veiller sur nous tous, pour sa gentillesse et sa bonne humeur. Un grand merci à tous les chercheurs du CMM, Beatriz Marcotegui, Jesús Angulo, Michel Bilodeau, Mathieu Faessel, Serge Beucher, Estelle Paradis, François Willot et Petr Matula pour m'avoir prodigué conseils, assistances et formations. Ce fût si enrichissant de travailler à leurs côtés.

---

Les collaborations industrielles m'ont amené à côtoyer de nombreuses personnes, tout au long de cette thèse, que je souhaite à présent remercier. Je remercie tous les collaborateurs du projet TOCATA, en particulier l'équipe de M. Leonetti : Julien Decreton, Mickaël Kolomytzeff, Franck Rein et Romain Casserat. Mes remerciements vont ensuite du côté d'Aulnay (L'Oréal) : merci à Thérèse Baldeweck, Anna-Maria Pena, Emanuelle Tancrède-Bohin et Marjorie Carpentier-Cheraud pour leur bonne humeur communicative et leur expertise en dermatologie. Enfin, merci à Jean Luc Gautier et Eric Coquelle, de l'entreprise Colas, pour avoir transmis leur expérience en défectologie des routes.

Je tiens à remercier Jean Philippe Vert, directeur du CBIO (Mines ParisTech), pour son expertise en analyse statistique lors de la conception de la méthode AdaCOS. Son aide a été précieuse.

J'ai également une pensée envers Jean-Marie Becker et Catherine Mennessier (CPE Lyon), qui ont su me transmettre leurs passions des mathématiques et du traitement d'images. Ils tiennent un rôle prépondérant dans mon parcours professionnel. Il en est de même pour Maxime Moreaud (IFP énergies nouvelles) qui m'a finalement convaincu de continuer pour trois années supplémentaires. Je les remercie tous les trois très sincèrement.

Si ces trois années à Fontainebleau ont été aussi agréables, c'est aussi grâce à mes frères de convolutions, thésard et post-doc : Jorge, Ali, Charles, Louise, Bruno, Jean, Guillaume, Luc, Nelly, Serge, Santiago, Julie, Hellen, Xiwei, Andrès, John, André, El Hadji, Torben, Vaia, Dario, Bart. Merci pour toutes les discussions, les souvenirs (et bières) partagés ensemble et surtout pour toutes ces soirées jeux endiablées (soirée jeux : où comment monter les stratégies les plus machiavéliques possible pour écraser l'adversaire dans un archipel bien connu de la mer Égée...).

Pour terminer, je remercie tous mes proches en particulier ma famille et ma belle famille qui m'ont témoigné une grande affection. Je pense particulièrement à mes parents, pour leur soutien sans faille tout au long de mon parcours et à qui je souhaite dire un éternel merci. Ils m'ont permis de me réaliser pleinement, aussi bien sur un plan personnel, que professionnel. J'ai aussi une pensée affectueuse envers mon frère et sa famille et envers mes grands parents. Je pense souvent à eux.

Enfin, je réserve ces derniers mots à celle qui partage ma vie, m'apaise, me comble et qui est devenue ma pacsette au cours de cette thèse. Virginie, ton amour est ce que j'ai de plus précieux. Merci pour tout !

Fontainebleau, le 22 octobre 2012



Vincent Morard

<b>Introduction</b>	<b>1</b>
Cadre des projets industriels . . . . .	1
Besoins et contraintes industriels . . . . .	3
Plan de cette thèse . . . . .	3
 <b>I Outils efficaces et rapides pour la détection de structures fines dans une image</b>	 <b>5</b>
<b>1 État de l'art : Détection des structures fines</b>	<b>7</b>
1.1 Banc de filtres directionnels . . . . .	8
1.2 Autres méthodes . . . . .	12
1.3 Bilan de l'état de l'art . . . . .	14
 <b>2 Élément structurant invariant par translation : nouvel algorithme mono dimensionnel pour calculer rapidement des ouvertures, des granulométries et des arbres de composantes connexes en temps constant par pixel</b>	 <b>17</b>
2.1 État de l'art . . . . .	18
2.2 Notions de bases sur la morphologie mathématique . . . . .	20
2.3 Nouvel algorithme pour des signaux 1-D . . . . .	23
2.4 Extension aux signaux 2-D . . . . .	29
2.5 Complexité . . . . .	33
2.6 Temps d'exécution et comparaison avec l'état de l'art . . . . .	34
2.7 Parallélisation de l'algorithme . . . . .	40
2.8 Bilan et perspectives . . . . .	41
 <b>3 Filtre par chemins : nouvel opérateur permettant de calculer des ouvertures parcimonieuses par chemins</b>	 <b>43</b>
3.1 État de l'art . . . . .	44
3.2 Notions de base sur les ouvertures par chemins . . . . .	44
3.3 Ouvertures parcimonieuses par chemins . . . . .	47

## TABLE DES MATIÈRES

---

3.4	Ouvertures parcimonieuses pour des chemins incomplets . . . . .	52
3.5	Considérations pratiques . . . . .	55
3.6	Précision des ouvertures parcimonieuses par chemins . . . . .	56
3.7	Résultats : applications industrielles . . . . .	66
3.8	Complexité algorithmique et temps de calcul . . . . .	67
3.9	Bilan et perspectives . . . . .	71
4	<b>Éléments structurants adaptatifs : nouveaux éléments structurants adaptatifs basés sur un processus de croissance de région et nouveaux opérateurs qui analysent leurs formes</b>	<b>73</b>
4.1	État de l'art . . . . .	74
4.2	Region growing structuring elements . . . . .	75
4.3	Analyse de la forme des REGSEs . . . . .	81
4.4	Résultats . . . . .	83
4.5	Considérations pratiques . . . . .	88
4.6	Bilan et perspectives . . . . .	94
5	<b>Filtres connexes : nouveaux filtres permettant de calculer des amincissements par attributs géodésiques et géométriques</b>	<b>97</b>
5.1	État de l'art . . . . .	98
5.2	Rappel : amincissement par attributs . . . . .	99
5.3	Attributs géodésiques . . . . .	101
5.4	Un nouvel attribut géodésique : le diamètre barycentrique . . . . .	104
5.5	Attributs géométriques . . . . .	109
5.6	Amincissement par attributs géodésiques et géométriques . . . . .	112
5.7	Résultats . . . . .	114
5.8	Algorithme et considérations pratiques . . . . .	116
5.9	Bilan et perspectives . . . . .	122
6	<b>Analyse statistique parcimonieuse : AdaCOS, nouvelle méthode de classification qui minimise une fonction de coût lors de la génération d'un modèle.</b>	<b>125</b>
6.1	Introduction et motivations . . . . .	126
6.2	Nouvelle méthode de classification : AdaCOS . . . . .	130
6.3	Nouvelle méthode de classification parcimonieuse : Forward AdaCOS . . . . .	131
6.4	Considérations pratiques et généralisation . . . . .	135
6.5	Résultats de classement . . . . .	136
6.6	Bilan et perspectives . . . . .	137
II	<b>Applications industrielles</b>	<b>139</b>
7	<b>Projet TOCATA : Extraction automatique des défauts de surface de pièces métalliques par traitement d'images et analyse statistique</b>	<b>141</b>

<b>8</b>	<b>Projet L'Oréal : Caractérisation 3-D <i>in vivo</i> du milieu fibreux du derme de la peau humaine par traitement d'images</b>	<b>143</b>
8.1	Présentation du projet L'Oréal . . . . .	144
8.2	Prétraitements . . . . .	146
8.3	Caractérisation du réseau de fibres . . . . .	149
8.4	Analyse statistique . . . . .	155
8.5	Interprétation des résultats . . . . .	159
8.6	Bilan et perspectives . . . . .	163
<b>9</b>	<b>Projet Colas : Contrôle non destructif des arrachements des routes par le traitement d'images et analyse statistique</b>	<b>165</b>
9.1	État de l'art . . . . .	166
9.2	Présentation du projet Colas . . . . .	166
9.3	Détection de la région d'intérêt . . . . .	168
9.4	Descripteurs de texture . . . . .	169
9.5	Apprentissage non supervisé . . . . .	174
9.6	Apprentissage supervisé . . . . .	175
9.7	Solution logicielle . . . . .	177
9.8	Bilan et perspectives . . . . .	181
	<b>Conclusions et perspectives</b>	<b>183</b>
	Contributions de cette thèse . . . . .	183
	Perspectives . . . . .	185
<b>III</b>	<b>Annexes confidentielles</b>	<b>187</b>



## TABLE DES MATIÈRES

---

« *The current state of knowledge can be summarised thus : in the beginning there was nothing, which exploded.* »

Terry Pratchett, Théorie du “Big Bang”

Le *traitement d’images* est un domaine vaste où les limites sont constamment repoussées pour apporter des solutions innovantes et adaptées aux besoins de la science et des industriels. Avec l’avènement des ordinateurs et des techniques d’imagerie, les champs applicatifs des méthodes d’analyse d’images ont explosé, allant de l’imagerie médicale et biomédicale aux contrôles non destructifs, en passant par la quantification de matériaux ou l’imagerie spatiale.

Les défis actuels en analyse d’images portent principalement sur la création de nouveaux traitements plus performants et plus rapides pour répondre à la fois à la complexification des problèmes et à l’évolution toujours croissante de la quantité de données à traiter. Aussi, l’association avec une discipline connexe et complémentaire, l’*analyse statistique*, émerge et devient indispensable pour de nombreuses applications.

Le travail présenté dans ce document de thèse est motivé par des besoins industriels pour la conception de nouveaux traitements permettant d’extraire et de rehausser **les éléments fins** d’une image d’une manière efficace et **rapide**. Ces outils seront couplés à des méthodes statistiques pour proposer des solutions originales aux industriels, partenaires du Centre de Morphologie Mathématique (CMM) : Safran, Dassault Aviation, Areva et d’autres petites et moyennes entreprises dans le cadre du projet TOCATA, L’Oréal pour une étude sur les fibres du derme de la peau et Colas pour un projet sur le contrôle non destructif des routes.

## Cadre des projets industriels

Cette thèse s’inscrit dans un contexte industriel fort : le projet TOCATA, d’une part, et deux projets réalisés pour le compte des sociétés L’Oréal et Colas, d’autre part. Ces trois projets sont très différents l’un de l’autre mais possèdent un fil conducteur commun : le contrôle par l’image et plus précisément, **les contrôles non destructifs**.

### Projet TOCATA

Le projet TOCATA<sup>1</sup> a pour objectif de remplacer une technologie de contrôle non destructif très polluante, le ressuage, par un contrôle optique complètement automatisé et propre. Un robot, portant un système optique, réalise une trajectoire précise autour de la pièce à inspecter. Le flot d'images provenant du capteur doit être analysé afin de vérifier la conformité de la pièce avec le cahier des charges.

Par conséquent, l'objectif consiste à développer une méthodologie adaptée, associant le traitement d'images et l'analyse statistique, afin de localiser tous les défauts de surface des pièces métalliques. Une stratégie générique sera alors utilisée afin de permettre la détection de différents types de défauts sur différents types de pièces. Les données fournies par la caméra représentent plusieurs giga pixels de données par pièce et concevoir des algorithmes optimisés et rapides constitue un enjeu primordial dans la réussite de ce projet. La précision de la détection des défauts est un autre facteur tout aussi indispensable. Il faudra donc limiter l'apparition de faux positifs tout en s'assurant que tous les défauts soient bien détectés.

L'accent a été mis sur les défauts de types **fissures** présents sur les pièces métalliques, et sur **la vitesse de calcul**, qui reste un leitmotif dans l'ensemble de ce document.

### Projet L'Oréal

Le partenariat avec l'entreprise L'Oréal porte sur la caractérisation de la peau grâce à un système optique innovant et non invasif : *la microscopie multiphoton*. Des images 3-D de la peau humaine sont acquises sur 60 volontaires répartis en deux catégories, les personnes ayant moins de 25 ans et celles ayant plus de 60 ans.

L'objectif de ce projet est la caractérisation des fibres du derme grâce à des techniques d'analyse d'images. La forme, la taille, l'orientation et l'organisation spatiale des fibres sont des paramètres mesurés et quantifiés. Dans un second temps, une interprétation possible est proposée en s'aidant de méthodes statistiques. Nous verrons ainsi quels sont les paramètres qui évoluent avec le temps. Suivre l'évolution des fibres du derme fournit des informations précieuses pour la compréhension globale des modifications de la peau et pour visualiser l'effet anti-âge des molécules actives mises au point par L'Oréal.

### Projet Colas

Dans le cadre des contrôles non destructifs, l'entreprise Colas souhaite étudier l'évolution des routes avec le temps afin d'obtenir des informations essentielles sur le vieillissement des voies de circulation. Obtenir automatiquement un taux de dégradation des routes facilitera les prises de décisions afin de remplacer ou non les sections endommagées. Un système optique est donc fixé à un véhicule adapté, permettant d'acquérir des images avec une résolution de 1 mm par pixel.

L'objectif consiste à développer et utiliser des outils de traitement d'images permettant d'analyser les différentes textures des routes. Une analyse statistique complètera cette étude pour détecter et extraire les portions de routes arrachées.

---

1. TOCATA : Technologie Optique Couplée à l'Analyse Topologique Automatisée

## Besoins et contraintes industriels

On peut dégager des besoins communs de ces trois projets industriels : (i) la détection et l'extraction des éléments fins d'une image, (ii) des outils statistiques pour la classification ou le classement et (iii) des interfaces hommes machines adaptées. Cependant, ces besoins, fixés par ces projets industriels, sont liés à des contraintes qu'il faut prendre en compte : (i) les temps de calcul des opérateurs de traitement d'images doivent être optimisés, (ii) les stratégies proposées doivent être génériques et (iii) les problématiques soulevées par les différentes applications doivent être solutionnées.

### Détection des structures filiformes

Euclide, mathématicien de la Grèce antique, a défini géométriquement la notion de courbe et de ligne comme un élément "*qui a une longueur sans largeur*". Cette phrase illustre parfaitement la principale caractéristique des structures filiformes ; une dimension de l'objet est très largement supérieure à l'autre. La détection de telles structures est de ce fait difficile, malgré tous les progrès réalisés par les systèmes optiques en termes de précision et de résolution.

L'objectif est d'extraire ces structures des images en réduisant le bruit et en supprimant les objets qui ne rentrent pas dans les spécifications. Il faut donc faire du filtrage et du rehaussement des objets filiformes pour aboutir à une partition (segmentation) de l'image en deux catégories distinctes : *les objets d'intérêt* et le *fond de l'image*.

En traitement d'images, de nombreuses méthodes existent pour extraire les éléments filiformes des images, cependant, toutes ne répondent pas aux contraintes soulevées par les projets industriels. Par conséquent, de nouvelles méthodes doivent être conçues pour améliorer soit la vitesse de calcul, soit la précision des opérateurs ou soit la généricité de la détection en proposant des méthodes pouvant extraire des structures filiformes ayant une tortuosité quelconque.

### Outils d'analyse statistiques

L'analyse statistique (ou la fouille de données) est une discipline transversale très utilisée et souvent indispensable de nos jours ; les systèmes étant de plus en plus complexes et de plus en plus gros : "*We are drowning in information and starving for knowledge*" - Roger Rutherford. Les méthodes d'analyse statistique permettent d'extraire l'information pertinente, de regrouper les éléments semblables, d'établir des règles et des liens entre les éléments, tout ceci dans le but d'*apprendre des données*.

L'analyse statistique a une place centrale dans ce document de thèse et l'associer au traitement d'images permettra de proposer des solutions à la fois précises et génériques. L'utilisateur pourra ainsi utiliser ces méthodes pour différents types de données moyennant une étape d'apprentissage statistique. On voit ici l'importance de proposer des interfaces graphiques simples, intuitives et adaptées pour qu'un utilisateur, sans connaissances particulières en traitement d'images ou en analyse statistique, puisse utiliser le logiciel.

## Plan de cette thèse

Ce document de thèse est décomposé en deux parties majeures : la première regroupe les apports méthodologiques axés sur la théorie de la **morphologie mathématique**, avec l'introduction de nouveaux opérateurs de traitement d'images pour la détection des éléments fins et d'une nouvelle méthode

d'analyse statistique. Enfin, la seconde partie présentera le travail réalisé spécifiquement pour chaque partenaire industriel dont les besoins ont été introduits brièvement dans la section précédente.

### Première partie : apports méthodologiques

Le chapitre 1 présente un état de l'art des différentes méthodes existantes permettant de rehausser ou de filtrer les structures filiformes. Suivent l'introduction et la description d'un nouvel algorithme permettant de calculer rapidement des ouvertures avec des éléments structurants rectilignes orientés dans toutes les directions de l'espace (chapitre 2). Une extension de cet algorithme permet aussi de construire des granulométries, des ouvertures par reconstruction et l'arbre de composantes connexes.

Cet algorithme constitue le socle du chapitre 3, où est défini un nouvel opérateur : *les ouvertures parcimonieuses par chemins*. Ce filtre permet de supprimer les chemins ayant une longueur inférieure à un paramètre. Les résultats sont sensiblement les mêmes que les ouvertures par chemins de Heijmans *et al.* [90] mais seulement un sous ensemble de chemins de l'image est analysé, permettant d'obtenir des gains considérables sur les temps de calcul et sur la précision de ces opérateurs.

Les chapitres 4 et 5 proposent des méthodes originales permettant d'extraire des structures filiformes très tortueuses de l'image. Le chapitre 4 est dédié à un opérateur supplémentaire relevant de la morphologie mathématique adaptative, les *region growing structuring elements*. Les éléments structurants créés ont une taille fixe mais une forme variable afin de s'adapter aux structures de l'image. De nouveaux opérateurs analysant la forme des éléments structurants adaptatifs sont présentés, permettant de rehausser les structures filiformes de l'image.

Quant au chapitre 5, il présente un opérateur connexe qui analyse la longueur des objets avec des mesures géodésiques ou géométriques. Ce filtre permet de supprimer les objets des images avec des critères de taille, d'élongation, de tortuosité ou de circularité. Une implémentation rapide est présentée, basée sur le calcul d'un nouvel attribut : le *diamètre barycentrique*.

Enfin, le chapitre 6 propose une nouvelle méthode d'analyse statistique appelée *AdaCOS*. Cette méthode est destinée au projet TOCATA et a pour objectif premier de réduire les temps de calcul. La méthode AdaCOS construit une régression linéaire pénalisée afin de réaliser un compromis entre la précision du modèle et la minimisation d'une fonction de coût. La généralisation de cette méthode permet d'obtenir des modèles parcimonieux et des régressions logistiques pénalisées.

### Seconde partie : applications industrielles

Les outils méthodologiques développés dans la première partie de ce document seront tous utilisés dans le cadre du projet TOCATA (chapitre 7). Ce chapitre présente les problématiques de ce projet et les solutions apportées pour la détection des défauts de surface de pièces métalliques.

Le chapitre 8 propose une application associée à la détection des éléments filiformes dans une image 3-D (projet L'Oréal). Les fibres de la peau sont caractérisées par des méthodes de traitement d'images suivies d'une analyse statistique pour extraire les paramètres importants.

Enfin, le chapitre 9 regroupe les travaux effectués pour l'entreprise Colas. Des outils pour analyser la texture des routes sont développés et des analyses statistiques non supervisées, puis supervisées, permettent de détecter et de quantifier les défauts de surface des routes.

## Première partie

# Outils efficaces et rapides pour la détection de structures fines dans une image



---

## État de l'art : Détection des structures fines

---

*« On ne connaît pas complètement une science tant qu'on n'en sait pas l'histoire. »*

Auguste Comte, Cours de philosophie positive, 1830-1842

En traitement d'images, la détection des structures filiformes se révèle être un problème épineux mais néanmoins indispensable dans un nombre important d'applications, que l'on peut répartir en plusieurs domaines d'étude :

1. **En imagerie médicale**, de telles structures peuvent être trouvées dans les vaisseaux sanguins, les vaisseaux du fond de l'œil, les molécules d'ADN, les os, les neurones, les empreintes digitales, les rides et les fibres de la peau, ou les fibres musculaires.
2. **Pour les contrôles non destructifs**, on trouve ces structures dans les défauts de surface de type fissure sur les routes ou sur des pièces métalliques, mais aussi pour le contrôle de la distribution de taille des fibres d'un matériau ou pour la détection des craquelures des tableaux anciens.
3. **Pour les images satellites ou aériennes**, on peut être amené à extraire la position des routes, des rails, des rivières ou des fleuves.

En observant les objets à détecter de ces applications, on peut dégager une caractéristique commune ; ils ont une forme fine spatialement tout en étant très allongés, rendant difficile leur détection. De plus, de telles structures ont une sensibilité accrue aux bruits, du fait de leur finesse et de leur longueur. Ces objets filiformes apparaissent donc souvent plus courts qu'ils ne le sont en réalité, biaisant la distribution de taille mesurée. Ainsi, les opérateurs de traitement d'images doivent détecter uniquement les objets filiformes tout en étant robustes aux bruits. De là vient la principale difficulté de ces méthodes : comment rehausser les structures fines de l'image sans augmenter le bruit ? Comment filtrer le bruit sans détruire l'information utile ?

### DÉFINITION 1.1

**Filtre** : le principe du filtrage consiste à modifier l'image pour en retenir exclusivement l'information pertinente.

On recense, dans ce chapitre, quelques méthodes utilisées dans la littérature pour détecter les structures fines d'une image. Ce chapitre n'a pas vocation à lister exhaustivement tous les opérateurs



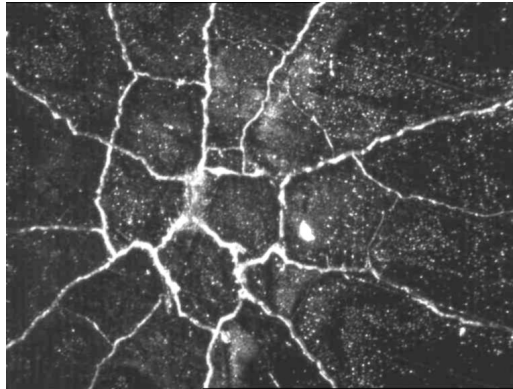


FIGURE 1.1 – Image initiale de taille 768 par 576 pixels, pour la détection des éléments fins de type fissures.

existants mais plutôt à extraire certains d'entre eux, qui présentent des caractéristiques intéressantes par rapport aux applications visées et à montrer leurs résultats en images. Voir la thèse de Tankyevych [225] pour un état de l'art plus complet dans ce domaine.

Nous étudierons tout d'abord les méthodes basées sur les bancs de filtres directionnels avec des filtres linéaires ou morphologiques. Puis, nous verrons d'autres méthodes basées sur la transformée de Hough, sur les lignes de crêtes ou sur la dérivée seconde. Nous introduirons ces opérateurs d'une manière succincte et non formelle afin de visualiser leur intérêt à travers un exemple concret d'utilisation : l'extraction d'une fissure provenant d'une image d'une pièce métallique. Ces défauts sont localement rectilignes et déconnectés par endroit (voir la figure 1.1).

### 1.1 Banc de filtres directionnels

Les méthodes utilisant un banc de filtres directionnels utilisent généralement un élément de base, appelé noyau pour les filtres linéaires ou élément structurant (ES) pour les filtres morphologiques (non linéaires), qui pourra subir des transformations affines comme des rotations ou des mises à l'échelle. Il s'agit donc de créer un noyau ou un ES de forme et d'orientation telle que la réponse soit maximale pour les objets que l'on souhaite détecter. Concrètement, ce noyau ou cet ES est convolué avec l'image et les résultats de ces filtres directionnels sont combinés ensemble en prenant le supremum des réponses. On obtient donc un filtre isotrope que l'on peut décliner en plusieurs catégories, les filtres construits avec des segments de droites, les filtres linéaires et les filtres avec des chemins.

#### 1.1.1 Segment de droite

Si les éléments à détecter dans l'image sont localement linéaires, alors utiliser un segment de droite comme noyau ou comme ES, orienté parallèlement à ces structures, permettra de les rehausser. On va alors déplacer ce segment de droite sur tous les pixels de l'image pour réaliser une convolution. Le pixel central aura une nouvelle valeur qui pourra être calculée de plusieurs façons différentes :

- Si on calcule la valeur moyenne des pixels du noyau et que l'on remplace la valeur du pixel central par cette nouvelle valeur, on obtient un filtre moyen directionnel.

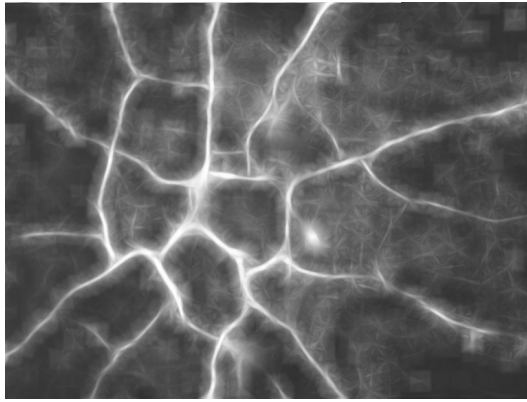
- En classant les pixels par ordre croissant et en sélectionnant la valeur médiane, on définit un filtre médian directionnel [236, 102].
- En attribuant au pixel central une combinaison linéaire entre un noyau de profil gaussien et l'image, on a un filtre gaussien directionnel.
- Pour préserver les contours de l'image, on peut pondérer le noyau de convolution non seulement par la distance spatiale mais aussi par la distance tonale. Cet opérateur, introduit par Tomasi et Manduchi [234], correspond à la généralisation du filtre gaussien, appelé filtre bilatéral directionnel.
- En calculant la succession d'une érosion et d'une dilatation, on a une ouverture morphologique directionnelle, [202, 215], opérateur qui sera défini formellement dans le chapitre 2.
- Enfin, pour rendre la détection moins sensible aux bruits poivre et sel de l'image, on peut utiliser des filtres de rang pour construire des opérateurs morphologiques (voir [209] pour une étude présentant l'état de l'art de ces méthodes). Plus précisément, Ronse [191, 192] puis Heijmans [88] présentèrent des ouvertures basées sur les filtres de rang (*rank-max openings*) qui permirent d'analyser rapidement toutes les combinaisons possibles en prenant exactement  $k$  pixels de cet élément structurant. Une ouverture de rang maximum est donc la composition d'un filtre de rang suivi d'une dilatation. On prendra ensuite l'infimum avec l'image initiale afin de conserver la propriété d'anti-extensivité de ce filtre.

On illustre le fonctionnement de ces six opérateurs sur le défaut de la figure 1.1. Les segments de droites sont créés avec des lignes de Bresenham [30], orientées suivant 180 directions dans l'espace. La figure 1.2 présente les résultats obtenus en calculant le supremum des filtres directionnels pour chaque opérateur. Pour un filtre moyen, on constate que l'on préserve et rehausse correctement les structures filiformes de l'image même si elles sont déconnectées par du bruit. Néanmoins, ce filtrage introduit une sorte de halo autour de toutes les structures. Cet effet est quelque peu réduit lorsque l'on utilise un filtre gaussien puis fortement atténué pour un filtre bilatéral ou un filtre médian. En effet, la qualité de détection de ces deux opérateurs est bonne, tout en limitant les artéfacts dans le fond de l'image. Enfin, les opérateurs morphologiques n'introduisent pas d'artéfacts, mais les fissures ne sont pas pontées contrairement aux filtres linéaires. On note que le supremum des ouvertures directionnelles avec un filtre de rang est moins sensible aux bruits qu'une ouverture morphologique classique et il préserve par conséquent plus de structures dans l'image.

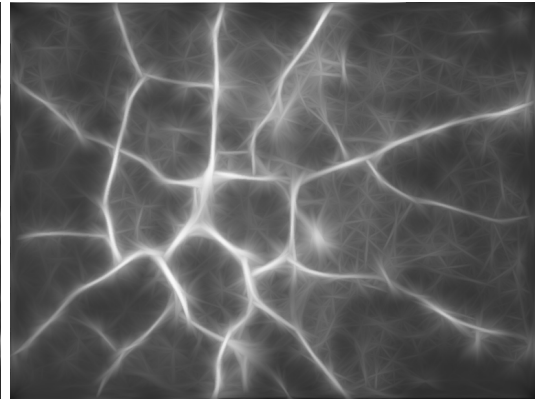
D'autres auteurs comme Sun et Vallotton, dans [222] proposèrent d'analyser le profil des éléments fins avec des segments de droites orientés dans toutes les directions de l'espace. Ils détectent alors le lobe principal ainsi que tous les lobes secondaires de ce profil 1-D, puis ils les préservent ou non suivant un critère de contraste.

### 1.1.2 Les filtres linéaires

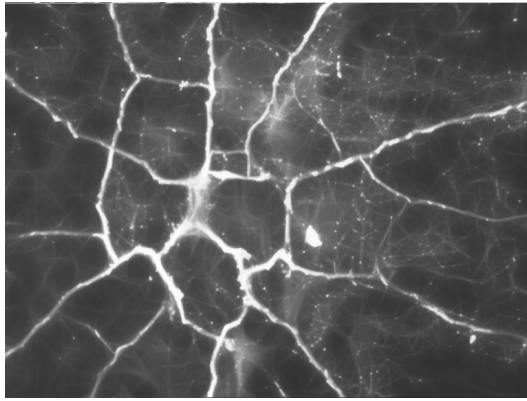
Mis à part le filtre bilatéral, ces filtres réalisent un traitement uniquement spatial. Cependant, l'information fréquentielle est tout aussi importante. Pour la prendre en compte, nous pouvons passer dans le domaine de Fourier, qui fournit une représentation purement fréquentielle de l'information [25]. A mi-chemin, les ondelettes permettent de réaliser une analyse à la fois spatiale et fréquentielle. Cette notion a été introduite pour la première fois au début des années 1900 dans le document de thèse de Haar [84] puis elle a été reprise en 1986 par Mallat [130] qui unifia les ondelettes dans le contexte du traitement du signal et de l'image. Enfin, Daubechies [47] utilisa le travail de Mallat pour construire un ensemble d'ondelettes ayant une base de fonctions orthogonales.



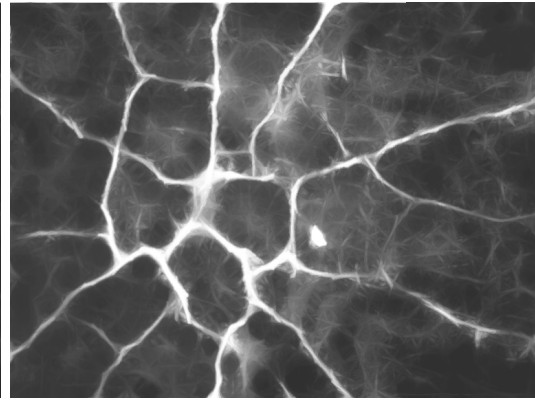
(a) Supremum des filtres moyens sur des segments de taille 31 pixels



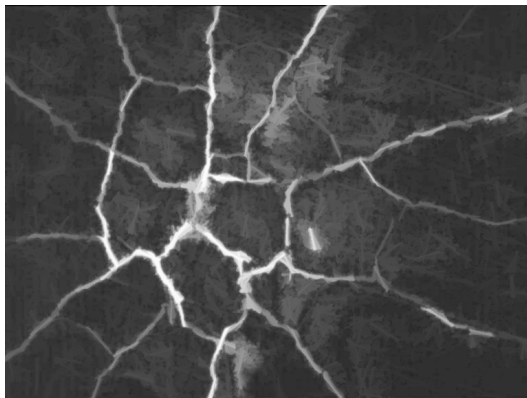
(b) Supremum des filtres gaussiens sur des segments de taille 31 pixels



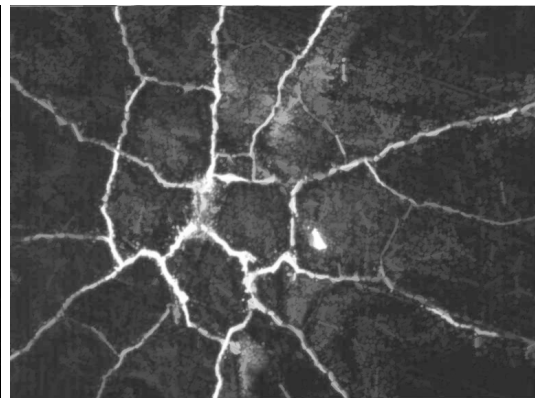
(c) Supremum des filtres bilatéraux sur des segments de taille 31 pixels



(d) Supremum des filtres médians sur des segments de taille 31 pixels



(e) Supremum des ouvertures par des segments de taille 31 pixels



(f) Supremum des ouvertures de rang maximal par des segments de taille 31 pixels avec un pourcentage de 80% pour le filtre de rang

FIGURE 1.2 – Illustration des filtres utilisant des segments de lignes comme éléments structurants ou comme noyaux. Chaque opérateur correspond au supremum des réponses de 180 filtres directionnels.

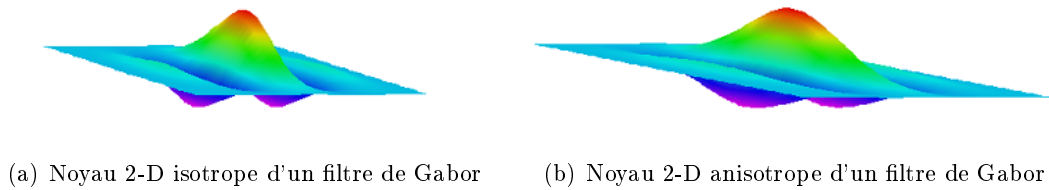


FIGURE 1.3 – Noyaux de convolution des filtres de Gabor : (a) dans sa version de base, (b) en supprimant l'isotropie de l'enveloppe gaussienne.

Ces opérateurs sont intrinsèquement multi-échelles et parmi toutes les méthodes existantes dans la famille des ondelettes, on en extrait deux, largement utilisées dans la littérature : les filtres de Gabor et les filtres *Difference of Gaussians* (DoG).

### Filtres de Gabor

Défini par Gabor en 1946 [70], ce filtre fait partie de la famille des ondelettes. Sa réponse impulsionnelle est constituée d'une enveloppe gaussienne, modulée par une sinusoïde plane. En modifiant l'orientation, la fréquence et la phase de la sinusoïde, on peut réaliser alors une analyse multi-échelle de l'image. Pour la détection des éléments filiformes d'une image, on rend l'enveloppe gaussienne anisotrope suivant la direction transverse à l'orientation de la sinusoïde. On obtient ainsi des noyaux de forme allongée qui s'adaptent bien aux objets filiformes. La figure 1.3 propose des représentations en trois dimensions de tels noyaux (isotropes et anisotropes) et la figure 1.5(a) présente un exemple d'utilisation de ce filtre. La détection des fissures est très bonne et la réduction du bruit efficace. On note néanmoins que l'on peut améliorer le traitement si on supprime les oscillations introduites par les lobes secondaires [232].

### Filtres “Difference Of Gaussians” (DoG)

Ce filtre défini par Marr et Hildreth dans [132] permet d'obtenir une approximation de la dérivée d'une gaussienne : un Lapacien. Ce filtre passe bande, intrinsèquement multi-échelles, réalise une convolution de l'image avec un noyau ayant une section qui correspond à la différence de deux gaussiennes avec deux écarts types différents. On peut modifier quelque peu la forme du noyau 2-D pour qu'il ait un profil Laplacien uniquement dans une direction, associée à un profil gaussien dans la direction transverse. On transforme ainsi cet opérateur en filtre directionnel. La figure 1.4(b) montre la construction d'un noyau DoG pour un signal 1-D ainsi qu'une représentation de ce noyau en trois dimensions.

La figure 1.5(b) présente le supremum des réponses avec un banc de filtres DoG directionnels pour la détection des éléments fins des images.

#### 1.1.3 Filtres avec des chemins

Ces noyaux de convolution permettent de détecter des structures fines mais linéaires. Travailler sur des chemins améliore la flexibilité pour pouvoir extraire des structures plus tortueuses.

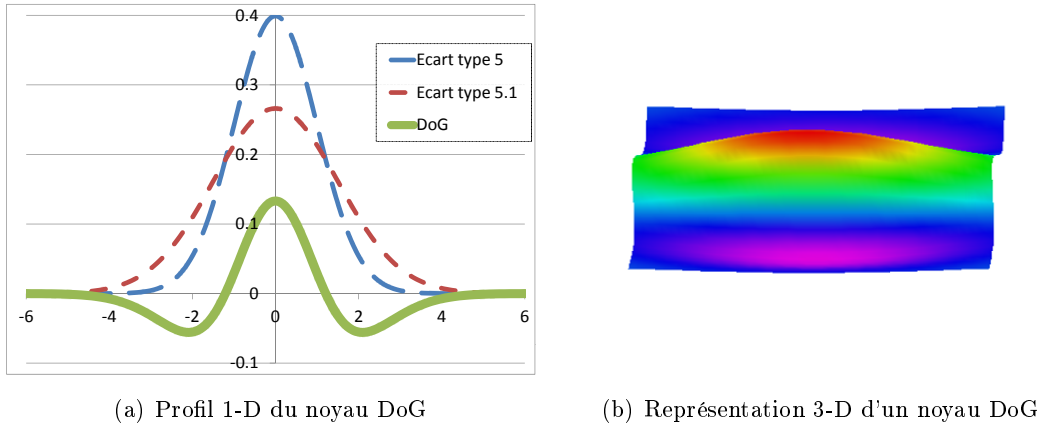


FIGURE 1.4 – Illustration du noyau de convolution utilisé pour la détection des structures fines et linéaires.

### Ouvertures par chemins

Vincent proposa dans [245] de trouver le chemin globalement minimal entre deux bordures de l'image. Cet opérateur fonctionne dans des environnements très bruités mais nécessite d'introduire de l'information à priori, pas toujours réalisable en pratique. Dans le même esprit, Heijmans *et al.* ont introduit en 2005 un opérateur morphologique très performant pour extraire les structures qui ne sont pas parfaitement linéaires. Cet opérateur, appelé *ouverture par chemins* [90], sera décrit précisément dans le chapitre 3, où l'on présente les *ouvertures parcimonieuses par chemins*, opérateur qui permet de réduire les temps de calcul tout en préservant la qualité de détection. On illustre par la figure 1.5(c), les résultats du supremum des ouvertures avec des chemins incomplets.

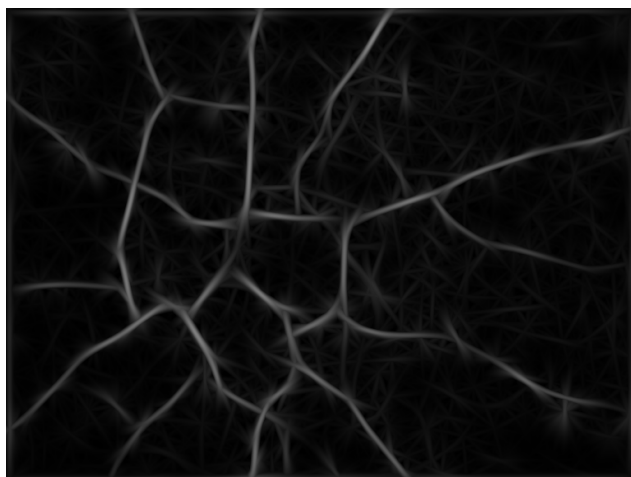
### Free form anisotropy

Nguyen *et al.* [169] proposent eux aussi une méthode utilisant des chemins, *free-form anisotropy*. Cet opérateur est dérivé des travaux de Roli [189] et analyse quatre chemins de taille fixe, construit d'une manière optimale autour du pixel central. Si un des chemins se situe sur une fissure, alors il sera statistiquement différent des autres, permettant ainsi de l'extraire. La figure 1.5(d) présente le résultat de ce filtre et on note que les structures fines sont correctement détectées; cependant des artéfacts apparaissent dans le fond de l'image.

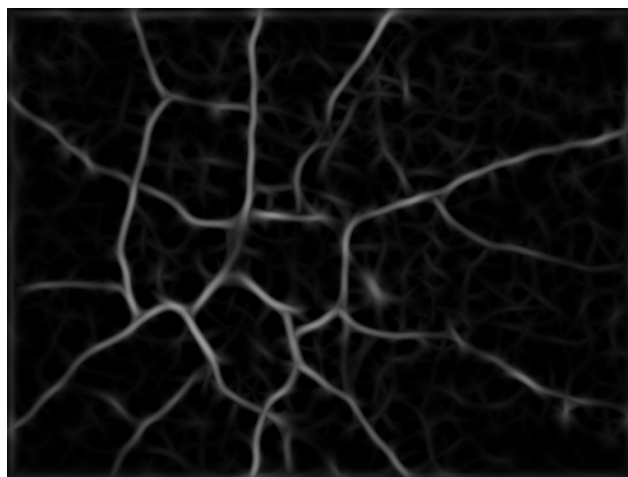
On peut citer d'autres algorithmes utilisant des bancs de filtres directionnels : des méthodes basées sur des modèles de vaisseaux sanguins : [67, 205, 235] ou des filtres pour la détection des routes à partir d'images satellites : [73] et [218].

## 1.2 Autres méthodes

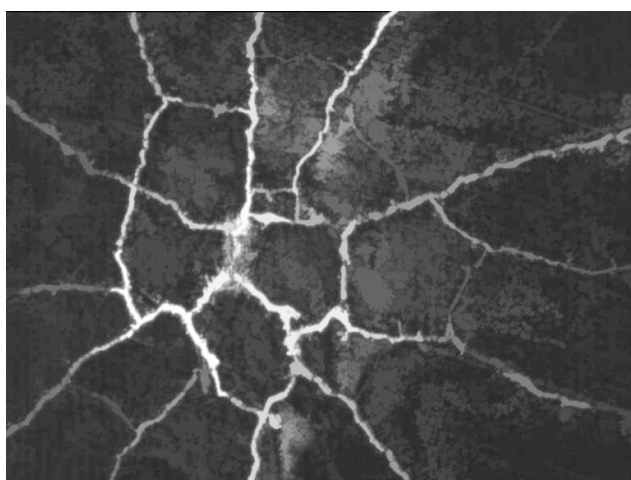
Il existe d'autres filtres et d'autres méthodes permettant de détecter les structures filiformes d'une image. On regroupe dans cette section quelques exemples d'opérateurs.



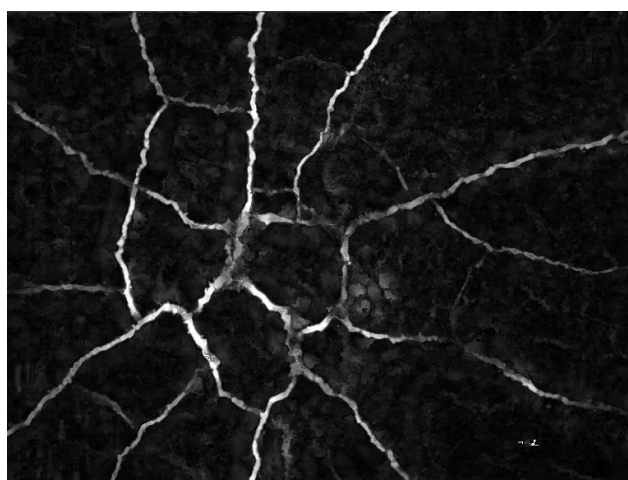
(a) Supremum des filtres de Gabor directionnels



(b) Supremum des filtres DoG directionnels



(c) Supremum des ouvertures par chemins incomplets



(d) *Free form anisotropy*

FIGURE 1.5 – Illustration des filtres de Gabor (a) et DoG (b) ainsi que les ouvertures par chemins (c) et les *free form anisotropy* (d).

### 1.2.1 La transformée de Hough et la transformée de Radon

Hough proposa en 1962, une transformation qui porte aujourd'hui son nom, permettant de détecter les lignes droites d'une image binaire [96, 56]. Il s'agit d'un changement d'espace de représentation où chaque point de l'image correspond à une sinusoïde dans l'espace des paramètres. Ainsi, la recherche de droites dans l'image se transforme en une recherche d'accumulation de points dans l'espace des paramètres. La transformée de Radon [186], propose une formalisation quelque peu différente mais il a été montré que ces deux transformations sont très proches. En effet, la transformée de Hough peut être vue comme une discrétisation de la transformée de Radon [220].

### 1.2.2 Lignes de crêtes, squelettes et lignes de partage des eaux

Les objets filiformes à détecter peuvent être vus comme des lignes de crêtes d'un relief topographique. L'opérateur de la ligne de partage des eaux [18, 247, 17] (LPE) permet de réaliser cette détection. Si on calcule une LPE sur l'image de la figure 1.5(a), en partant de tous ses h-minima<sup>1</sup>, on obtient alors une ligne de partage des eaux fine, qui colle parfaitement aux lignes de crêtes de l'image.

On peut aussi calculer un squelette homotopique d'une image en niveaux de gris afin de correctement positionner le squelette sur les lignes de crête de l'image [55]. Dans [190], Ronse proposa d'utiliser des noyaux de convolution adaptés pour détecter les lignes de crêtes afin de positionner un squelette.

### 1.2.3 Méthodes basées sur la dérivée

La détection des contours [132, 32, 51, 123] regroupe de nombreuses méthodes permettant de détecter les bords des structures filiformes. Le filtre de Canny [32] utilise l'information générée par la dérivée seconde d'un noyau gaussien afin de détecter la géométrie locale des structures. C'est un filtre optimal au sens des trois critères suivants : les contours sont bien détectés, bien localisés et fins.

Dans [223], Talbot travailla sur la recherche de segment de droite en combinant des informations locales et non locales. La recherche des lignes est réalisée depuis les maxima locaux de l'image en liant ces points par des méthodes de propagations et de reconnections.

En 2008, Jeulin et Moreaud [100] estimèrent l'orientation locale de fibres par l'analyse de la matrice de covariance du gradient de l'image. L'implémentation de cette méthode peut être réalisée dans le domaine de Fourier pour diminuer les temps de calcul.

Altendorf et Jeulin, [3, 1], utilisèrent des cartes de distances directionnelles associées à l'analyse de l'inertie pour définir les caractéristiques locales d'un réseau de fibres (orientation et rayon). En dimension 2, quatre directions sont nécessaires pour estimer les paramètres locaux afin d'obtenir une implémentation rapide.

L'analyse de la matrice Hessienne peut donner de l'information sur l'orientation locale des structures de l'image. Cette stratégie a été utilisée par Tankyevych *et al.* dans [226] pour filtrer les structures en fonction de leur orientations locales.

## 1.3 Bilan de l'état de l'art

Les méthodes présentées dans ce chapitre permettent d'extraire ou de rehausser les structures filiformes d'une image. Cependant, pour quantifier la qualité de la détection proposée par ces opérateurs, plusieurs facteurs entrent en jeux :

---

1. Un filtre h-minima supprime tous les maxima locaux de l'image qui ont une hauteur inférieure à  $h$ .

1. Un des principaux facteurs réside dans la précision des opérateurs puisqu'il est nécessaire d'extraire uniquement les structures filiformes de l'image. Dans une sous catégorie, un opérateur sera encore plus précis lorsqu'il pourra trier ces structures par rapport à leur longueur, à leur largeur ou à leur élongation.
2. Un opérateur doit être robuste au bruit pour ne pas sous-estimer la taille des structures filiformes qui sont déconnectées par du bruit.
3. Un opérateur doit pouvoir détecter des structures qui ont une tortuosité quelconque.
4. Enfin, un dernier facteur, essentiel pour notre étude, consiste à avoir une détection très rapide pour répondre aux fortes contraintes temporelles des applications industrielles.

Il est évident qu'un unique opérateur ne peut pas avoir toutes ces qualités en même temps. Certaines méthodes permettent d'extraire uniquement les structures rectilignes, d'autres travaillent avec des chemins et peuvent extraire des structures plus tortueuses. De plus, la précision de ces méthodes n'est pas équivalente, tout comme les temps de calcul qui peuvent varier de plusieurs ordres de grandeur. Pour de nombreux opérateurs, il faut faire un compromis entre temps de calcul et précision.

Parmi l'ensemble des méthodes disponibles dans la littérature, on remarque que peu d'entre elles ont des temps de calcul satisfaisants pour nos applications industrielles. Ceci constituera notre premier axe de travail. Nous proposerons donc des algorithmes rapides, permettant d'accélérer des opérateurs existants (chapitre 2) ou des nouveaux opérateurs précis et rapides : les ouvertures parcimonieuses par chemins, chapitre 3.

Le deuxième axe de recherche consiste à créer des filtres permettant d'extraire des structures filiformes qui ont une forte tortuosité (chapitre 4 et 5). En effet, la majorité des méthodes est conçue pour la détection des structures plus ou moins rectilignes.

Enfin, on propose un troisième axe de recherche original qui associe l'analyse statistique et le traitement d'images pour extraire les éléments filiformes d'une manière rapide et efficace (le projet TOCATA, chapitre 7, utilise la méthode d'analyse statistique développée dans le chapitre 6).



## 1. ÉTAT DE L'ART

---

---

### Élément structurant invariant par translation : nouvel algorithme mono dimensionnel pour calculer rapidement des ouvertures, des granulométries et des arbres de composantes connexes en temps constant par pixel

---

Dans un contexte industriel, le temps de calcul est souvent un facteur déterminant. Élaborer des algorithmes rapides permet d’appréhender des problèmes qui ont des fortes contraintes temps réel. Dans ce chapitre nous présentons un nouvel algorithme permettant de calculer des ouvertures morphologiques, des ouvertures par reconstruction, des granulométries ainsi que des arbres de composantes connexes le long de signaux mono dimensionnels (1-D). Il s’agit d’un algorithme simple, rapide et flexible qui ne nécessite en tout et pour tout, qu’un unique balayage du signal. La complexité théorique est linéaire par rapport au nombre d’éléments du signal et constante par pixel. De plus, cet algorithme peut travailler *in situ*<sup>1</sup> et gérer n’importe quel type de données du moment que les valeurs appartiennent à un groupe ordonné. En pratique, nous utilisons des entiers sur 8-bits, 16-bits ou 32-bits et des flottants simples ou doubles précision. De plus, la flexibilité de cet algorithme permet d’adapter la gestion des bords aux différentes applications.

Pour étendre ces opérateurs à des images à deux dimensions (2-D) ou plus, deux extensions sont proposées : (i) on extrait des droites 1-D avec des orientations arbitraires. On peut donc définir et utiliser des granulométries orientées pour caractériser finement les textures, ainsi que des ouvertures directionnelles pour conserver les structures fines et rectilignes dans une image. (ii) En définissant un balayage original sur des chemins, on réalise un filtrage des structures fines mais qui ne sont pas nécessairement droites. Un nouvel opérateur sera proposé dans le chapitre 3, *les ouvertures parcimonieuses par chemins*.

On montrera, à travers de nombreux benchmarks (tests de performance) que les temps de calcul de cet algorithme sont :

1. presque indépendants de la taille de l’élément structurant (les temps de calcul diminuent avec la taille de l’ES) ;
2. les plus rapides pour des ouvertures 1-D avec des données d’entrée ayant une précision supérieure à 8-bits ;

---

1. *In situ* : l’image d’entrée peut être la même que l’image de sortie

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

---

3. compétitifs pour des données d'entrée 8 bits, par rapport à un algorithme introduit par Van Droogenbroeck et Buckley [239] ;
4. les plus rapides pour le calcul des granulométries.

Enfin, on notera que le nombre de pixels par seconde que l'on pourra traiter augmente avec la taille de l'image.

Mots clés : Morphologie mathématique, ouvertures linéaires, ouvertures linéaires par reconstruction, granulométries 1-D, arbre de composantes connexes 1-D, temps constant par pixel.

Ce travail a donné lieu à trois publications :

1. Dans [156], l'algorithme pour le calcul des ouvertures 1-D rapides est présenté pour la première fois à la conférence internationale ICASSP en 2011 :  
**V. Morard, P. Dokládál, and E. Decenciére, Linear openings in arbitrary orientation in  $O(1)$  per pixel, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011**
2. Dans Karas *et al.* [104], le calcul des ouvertures 1-D est déporté sur une carte graphique. Il s'agit d'un travail conjoint entre l'ESIEE Paris<sup>a</sup> et le Centre de Morphologie Mathématique, publié dans le *Journal of Real-Time Image Processing* :  
**P. Karas, V. Morard, J. Bartovsky, T. Grandpierre, E. Dokládlová, P. Matula, and P. Dokládál, GPU implementation of linear morphological openings with arbitrary angle, *Journal of Real-Time Image Processing*, 2012**
3. Dans Morard *et al.* [155], cet algorithme est étendu au calcul des granulométries et de l'arbre de composantes connexes et publié dans une édition spéciale consacrée aux filtres morphologiques dans le journal *Selected Topics in Signal Processing* (STSP) :  
**V. Morard, P. Dokládál, and E. Decenciére, One-dimensional openings, granulometries and component trees in  $O(1)$  per pixel, *IEEE Journal of Selected Topics in Images Processing*, 2012**

---

<sup>a</sup>. ESIEE Paris est un centre de formation supérieure scientifique et technique de la Chambre de Commerce et d'Industrie de Paris : <http://www.esiee.fr/>

### 2.1 État de l'art

Parmi l'ensemble des méthodes permettant d'extraire les éléments fins d'une image, le supremum des ouvertures par segments orientés dans toutes les directions est une méthode largement utilisée dans la littérature, qui répond en partie aux contraintes soulevées par nos applications. Il est cependant nécessaire de réduire au maximum les temps de calcul de cet opérateur pour pouvoir l'utiliser dans des applications qui ont de fortes contraintes temporelles (voir le chapitre 7).

Pour rappel, la morphologie mathématique (MM) est historiquement la première théorie non-linéaire dans le domaine du traitement des images. Inventée en 1964 par deux chercheurs français

de l'école des Mines de Paris : Matheron et Serra, la morphologie mathématique est basée sur une approche ensembliste. L'idée de base est de comparer un ensemble inconnu à un élément structurant (ES), qui lui, sera parfaitement caractérisé par sa forme et par sa taille. Cette comparaison donnera des informations sur la géométrie des objets que l'on souhaite analyser. Une introduction à la MM est donnée dans les ouvrages suivants : [137, 202, 203, 88, 210].

Historiquement, le calcul d'une ouverture nécessite la composition des deux opérateurs de base de la morphologie mathématique, l'érosion suivie de la dilatation [202]. La complexité d'un tel algorithme suivant rigoureusement cette définition est en  $\mathcal{O}(NL)$ , avec  $L$  le nombre de pixels de l'élément structurant et  $N$  le nombre de pixels de l'image. Par conséquent, les temps de calcul sont longs et dépendent de la taille de l'élément structurant. De nombreux chercheurs se sont penchés sur cette problématique et ont proposé des algorithmes ayant une complexité plus faible.

En 1985, Petch [180] remarqua la grande redondance des données entre deux éléments structurants de pixels voisins. Il proposa donc une décomposition logarithmique de cet ES ce qui permit de réduire considérablement les répétitions superflues. Cet algorithme a été repris puis amélioré par Coltuc et Pitras [40] pour réduire la complexité algorithmique à  $\mathcal{O}(N \log(L))$ . Plus tard, Van Herk en 1992 [241] ainsi que Gil et Werman en 1993 [77] proposèrent indépendamment un algorithme en  $\mathcal{O}(N)$ , indépendant de la taille de l'élément structurant. Cet algorithme calcule des érosions ou des dilatations en deux balayages du signal 1-D (un balayage avant et un arrière) et permet d'obtenir le résultat désiré avec seulement trois comparaisons par pixel en moyenne, quelque soit la taille de l'ES. Cet algorithme constitue un avancement majeur et il est toujours largement utilisé de nos jours. Depuis, de nombreux chercheurs reprirent et améliorèrent cet algorithme dans le but d'obtenir moins de comparaisons par pixel [147, 74, 76, 39]. Cependant les temps de calcul de ces algorithmes une fois implémentés, restent proches de ceux de l'algorithme original. Parmi eux, l'algorithme défini par Clienti *et al.* [39] en 2008, a permis de réduire la latence en supprimant le balayage arrière.

Enfin, une nouvelle classe d'algorithmes a été introduite récemment, permettant de calculer directement des ouvertures 1-D sans passer par le calcul de l'érosion puis de la dilatation. Van Droogenbroeck et Buckley [239] introduisirent un algorithme utilisant un histogramme et des ancras, permettant de calculer très rapidement des ouvertures 1-D pour des images 8-bits. Enfin, en 2011 Bartovsky *et al.* [13] proposèrent un algorithme adapté à des architectures type FPGA fonctionnant en *stream*<sup>1</sup>, pour calculer des ouvertures 1-D avec des orientations quelconques sur un support 2-D.

Une famille d'ouvertures peut être utilisée pour calculer des granulométries. Une granulométrie est un outil introduit initialement par Matheron [136] en 1967 pour analyser la distribution de tailles d'un matériau poreux. Pour comprendre son fonctionnement, on peut comparer cet outil à un processus de tamisage. Pour connaître la distribution de taille d'un échantillon granulaire, des tamis à mailles rectangulaires (ou d'une autre forme) et de tailles croissantes sont utilisés. L'analyse de la quantité de grains passant par chaque tamis, fournit la distribution de la taille des grains. Dans cette analogie, chaque tamis correspond à une ouverture, la forme des mailles correspond à la forme de l'élément structurant et calculer une granulométrie complète revient à filtrer l'image initiale avec des ouvertures de tailles croissantes. Un algorithme suivant strictement la définition de cet opérateur, donne des temps de calcul souvent prohibitifs pour de nombreuses applications. Pour des signaux 1-D, Vincent [246] développa un algorithme rapide analysant récursivement l'ensemble des maxima régionaux du signal. L'accélération obtenue est de plusieurs ordres de grandeur. Pour des signaux 2-D, Breen et Jones [29] proposèrent un algorithme connexe rapide facilitant l'utilisation intensive des granulométries

1. En stream : l'algorithme reçoit un flot continu de données qui doit être traité en une seule passe sur l'image avec peu de mémoire disponible

dans de nombreuses applications : distribution de taille, segmentation ou caractérisation de texture [173, 131, 229].

Une granulométrie est un opérateur multi-échelle qui analyse l'image pour différentes tailles. Dans [197], Salembier et Garrido introduisirent une structure de données synthétisant l'ensemble des données multi-échelles appelée *arbre de composantes connexes* ou *max tree*. Cette structure est utilisée dans la segmentation d'images ou de vidéos [197, 200, 101] dans la compression d'images [199] ou même dans la ligne de partage des eaux topologique [17, 44]. La construction de l'arbre de composantes connexes est coûteuse en temps mais des implémentations rapides existent : Najman et Couprie [163] décrivent un algorithme en temps quasi constant et Menotti *et al.* [142] en temps constant pour des signaux 1-D.

L'algorithme présenté dans ce chapitre permet de calculer d'une manière rapide et efficace des ouvertures, des granulométries ou des arbres de composantes connexes. Nous ferons tout d'abord quelques rappels sur les notions de base de la morphologie mathématique, section 2.2, puis nous décrirons ce nouvel algorithme pour des signaux 1-D, section 2.3, puis pour des images 2-D, section 2.4. Enfin, nous étudierons la complexité (section 2.5) et les temps de calcul de cet algorithme en le comparant avec l'état de l'art, section 2.6.

## 2.2 Notions de bases sur la morphologie mathématique

On rappelle dans cette section, les définitions des ouvertures par attributs et des granulométries pour des signaux 1-D et nous montrons comment les utiliser pour des images 2-D. Par convention, les opérateurs binaires (resp. en niveaux de gris) sont notés avec des majuscules (resp. en minuscules).

### 2.2.1 Ouvertures par attributs

Soit  $X : E \rightarrow \{0,1\}$  un signal binaire 1-D avec  $E$  un intervalle de  $\mathbb{Z}$ . On définit  $\{X_i\}$  comme l'ensemble de toutes les composantes connexes (CC) du signal, avec  $X_i$ , l'élément numéro  $i$  de cet ensemble. On remarque qu'en dimension 1 les CCs sont des intervalles de  $E$ . Ainsi, on souhaite filtrer les CCs en utilisant un critère associé à un attribut. Par exemple, si on prend comme attribut leur longueur, alors un critère possible sera "*la composante connexe sera supprimée si sa longueur est inférieure à un seuil donné*". Formellement, ce critère, noté  $\chi$ , est une application de  $\{X\}$  dans  $\{\text{faux}, \text{vrai}\}$  et permet de définir la fonction  $\Psi$  pour toutes les CCs du signal :

$$\Psi_\chi(X_i) = \begin{cases} X_i & \text{si } \chi(X_i) \text{ est vrai} \\ \emptyset & \text{sinon,} \end{cases} \quad (2.1)$$

#### DÉFINITION 2.1

L'ouverture par attribut du signal  $X$ , avec le critère  $\chi$  est donnée par l'équation suivante :

$$\Gamma_\chi(X) = \bigcup_i \Psi_\chi(X_i). \quad (2.2)$$

Dans ce chapitre, trois attributs seront retenus. Le premier sera la longueur de la composante connexe associée au critère "*supérieure au égale à  $L$* ". On notera  $\gamma_L$ , l'ouverture 1-D linéaire de taille  $L$  qui en découle. Le second attribut sera "*la CC contient un point d'un autre ensemble*", ce qui permettra de définir la reconstruction d'un signal marqueur sous le signal original. Enfin, le troisième attribut

utilisé sera la composition des deux précédents, “supérieure ou égale à  $L$  et contient un point d’un autre ensemble”. Ce critère nous permettra d’obtenir des ouvertures par reconstruction  $\gamma_L^{rec}$ .

Soit  $g$  une application de  $E$  dans  $V$ , avec  $V$  l’ensemble des niveaux de gris du signal 1-D,  $V = \mathbb{R}$  ou  $\mathbb{Z}$ . On définit  $X^h = \{x | g(x) \geq h\}$  comme l’image binaire obtenue après la binarisation de  $g$  au niveau  $h$ . On rappelle qu’une ouverture est anti-extensive, idempotente et croissante [202] et que, grâce à la propriété de croissance, l’ouverture commute avec l’opérateur binarisation. Ainsi, l’extension d’une ouverture binaire à des signaux en niveaux de gris est donnée par l’équation suivante :

$$\gamma_\chi(g)(x) = \vee \{h \in V \mid x \in \Gamma_\chi(X^h(g))\}, \quad (2.3)$$

avec le symbole  $\vee$ , le supremum.

Calculer des ouvertures sur des lignes prend tout son sens lorsque l’on étend cet opérateur à des images 2-D. Soit  $f : D \rightarrow V$  une image 2-D avec  $D \in \mathbb{Z}^2$ . On introduit un paramètre  $\alpha$  qui correspond à l’orientation des lignes que l’on va extraire à partir d’une image 2-D. La variable  $\alpha$  est orientée dans le sens trigonométrique et appartient à l’intervalle de valeurs  $[0^\circ, 180^\circ[$ . On obtient donc un opérateur directionnel noté  $\gamma_\chi^\alpha$ , qui nous permet de définir le supremum des ouvertures 1-D :

$$\vee \gamma_\chi(f) = \bigvee_{\alpha \in [0^\circ, 180^\circ[} \gamma_\chi^\alpha(f). \quad (2.4)$$

$\bigvee$  correspond au supremum de fonctions et cet opérateur est calculé en prenant le supremum des ouvertures orientées dans l’ensemble des directions de l’espace. Parallèlement au calcul du supremum des ouvertures 1-D, on peut définir en chaque pixel, l’angle de l’ouverture qui a donné la réponse maximale. On obtient ainsi l’opérateur  $\zeta_\chi(f)$ , permettant de définir l’orientation locale des structures de l’image :

$$\zeta_\chi(f) \in \underset{\alpha \in [0^\circ, 180^\circ[}{\operatorname{argsup}} \gamma_\chi^\alpha(f). \quad (2.5)$$

On note que plusieurs angles peuvent donner la même valeur pour l’ouverture. On choisira en pratique la première valeur de l’angle permettant d’obtenir la réponse maximale. La figure 2.1 regroupe les trois opérateurs,  $\gamma_\chi^\alpha$ ,  $\vee \gamma_\chi$  et  $\zeta_\chi$ , calculés sur une image 2-D d’empreintes digitales.

### 2.2.2 Distribution de tailles

Calculer une distribution de tailles des éléments présents dans une image, c’est analyser cette même image avec un ensemble d’ouvertures de tailles croissantes. Ce processus est aussi appelé granulométrie et il est défini par Matheron [137] de la manière suivante :

#### DÉFINITION 2.2

Une famille d’ouvertures  $(\gamma_v)_{v \geq 0}$ , forme une granulométrie si et seulement si :

$$\forall v \geq 0, \forall \mu \geq 0, v \geq \mu \Rightarrow \gamma_v \leq \gamma_\mu. \quad (2.6)$$

Plus tard, Maragos [131] fit l’analogie des granulométries avec le spectre de la transformée de Fourier et proposa une formulation de la distribution de tailles dans le domaine continu, notée  $PS$  (comme *Pattern Spectrum* en anglais) :

$$(PS(g))(v) = \frac{-d(Meas(\gamma_v(g)))}{dv}, \quad v > 0, \quad (2.7)$$

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

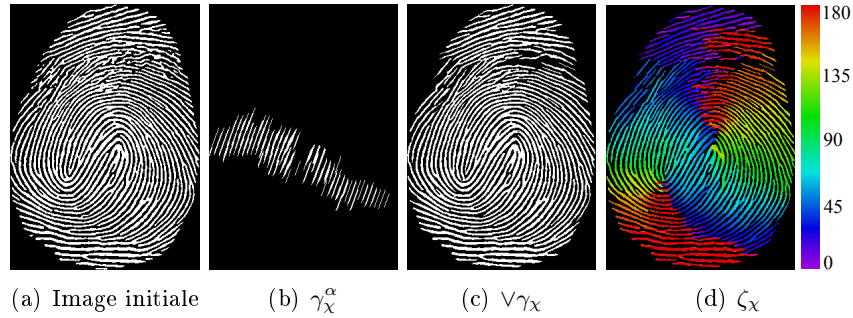


FIGURE 2.1 – Illustration des ouvertures 1-D utilisant le critère suivant : “les structures d’une longueur inférieure à 21 pixels sont supprimées.” (a) image initiale, une empreinte digitale. (b) ouverture directionnelle avec un angle  $\alpha = 70^\circ$ . Toutes les structures qui ont une longueur inférieure à 21 pixels et qui ne sont pas orientées suivant  $\alpha$  sont supprimées. (c) supremum des ouvertures dans toutes les directions. On préserve toutes les structures linéaires qui ont une taille suffisante. (d) détection de l’orientation locale en fausses couleurs.

ou  $Meas$  correspond à une mesure additive que l’on substituera par le volume dans le reste de ce chapitre. Dans le domaine discret, la différentielle d’ordre 1 est remplacée par la soustraction des résultats de deux ouvertures de tailles successives. Ainsi, l’analyse des résidus obtenus permet de connaître les éléments supprimés à cette échelle :

$$(PS(g))(v) = Meas(\gamma_v(g) - \gamma_{v-1}(g)), \quad v > 0. \quad (2.8)$$

On remarque que la variable  $PS$  est maintenant un histogramme où chaque classe correspond à une taille précise. La figure 2.2 présente un signal 1-D ainsi que sa contribution au  $PS$  associé. On prendra pour cet exemple, une famille d’ouvertures avec un attribut de longueur et comme mesure le volume. Par conséquent, le bloc  $c_3$ , ayant une longueur de 5 pixels et un volume égale à 15, ajoute 15 à la 5<sup>ème</sup> classe de l’histogramme. Les éléments  $c_4$  et  $c_6$  ont, quant à eux une longueur unitaire et contribuent donc à la première classe. Le même processus est appliqué à l’ensemble des blocs du signal.

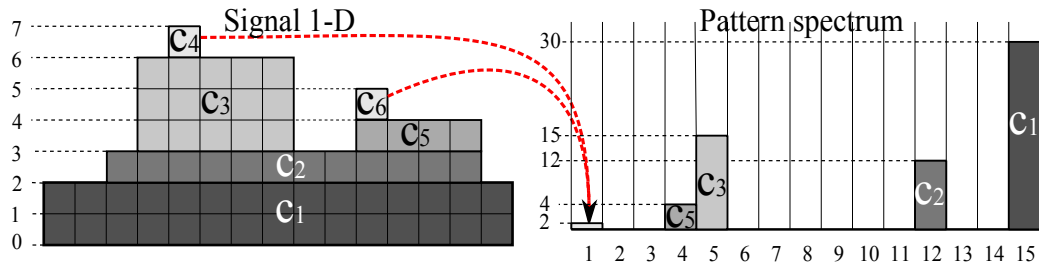


FIGURE 2.2 – Contribution d’un signal 1-D aux  $PS$  (distribution de taille)

La section suivante propose de calculer efficacement les ouvertures et les granulométries pour des signaux 1-D.

## 2.3 Nouvel algorithme pour des signaux 1-D

Cette section introduit un algorithme permettant de calculer un arbre de composantes connexes, une granulométrie ou une ouverture par attribut de longueur à partir d'un signal 1-D. Outre la flexibilité, la vitesse d'exécution de cet algorithme est un élément crucial qui sera précisément décrit et commenté dans cette section. À la base de cet algorithme rapide, on retrouve une décomposition compacte et efficace du signal 1-D en une structure appelée *cordes*.

### 2.3.1 Décomposition du signal en cordes maximales

On considère un signal 1-D,  $g : E \rightarrow V$ , avec  $V = \mathbb{R}$  ou  $\mathbb{Z}$ . On rappelle que  $X^h$  décrit la binarisation de  $g$  au niveau  $h$  et que  $\{X_i^h\}$  est l'ensemble des CCs à ce niveau  $h$  (on réindexe  $\{X_i^h\}$  en  $\{X_j\}$ ). On remarque qu'une même composante connexe peut être définie pour plusieurs seuils successifs. Néanmoins, on souhaite avoir une représentation de  $g$  la plus compacte possible et la réduction des doublons passe par trouver  $h$  le plus haut possible, permettant de définir et d'extraire une CC.

#### DÉFINITION 2.3

On appelle corde tous les couples  $c = (X_j, k)$  appartenant à  $\{X_j\} \times V$ , tel que :

$$k = \min_{x \in X_j} g(x), \quad (2.9)$$

où  $k$  correspond à l'altitude de la corde. Comme  $X_j$  est un intervalle de  $\mathbb{Z}$ , on peut l'écrire  $[sp, fp]$  avec  $sp$  et  $fp$  respectivement la position initiale et finale de la corde. Ainsi, la longueur d'une corde est égale à  $Length(c) = fp - sp + 1$ . La figure 2.3 (a) illustre la décomposition d'un signal 1-D en un ensemble de cordes :

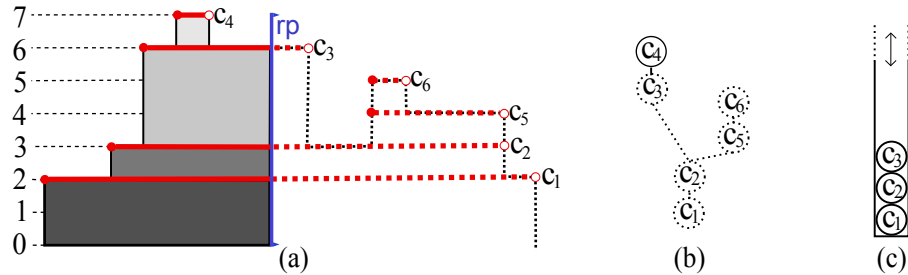


FIGURE 2.3 – (a) signal 1-D et les cordes associées. La lecture du signal se fait séquentiellement de la gauche vers la droite et la position de la tête de lecture est représentée par  $rp$  (les lignes continues (resp. en pointillées) représentent les éléments connus (resp. inconnus)) ; (b) arbre de composantes connexes ; (c) état courant de la pile (en fonction de  $rp$ ).

La longueur des cordes respecte la propriété d'inclusion. En effet, pour deux cordes  $c_i = \{X_i, k_i\}$  et  $c_j = \{X_j, k_j\}$  telles que  $X_i \subset X_j$ , alors on aura  $k_i > k_j$ . On dit alors que  $c_j$  est l'ancêtre de  $c_i$  et réciproquement,  $c_i$  est le descendant de  $c_j$ . Si on admet que  $c_i$  est le plus grand descendant de  $c_j$  au sens de la longueur de la corde, alors on dit que  $c_i$  est le *fils* de  $c_j$  ou que  $c_j$  est le *père* de  $c_i$ . Cette relation père-fils permet de définir l'arbre de composantes connexes ou max-tree ainsi que le volume



d'une corde  $c_i$  :

$$V(c_i) = (k_i - k_j)Length(c_i). \quad (2.10)$$

### DÉFINITION 2.4

L'ensemble des cordes d'un signal, noté  $C = \{(X_j, k_j)\}$ , est une représentation strictement équivalente au signal  $g$  et la reconstruction de ce signal au point  $x$ , est calculée en prenant l'altitude maximale de toutes les cordes de  $C$  contenant  $x$  :

$$g(x) = \max_{(X_j, k_j) \in C, x \in X_j} k_j. \quad (2.11)$$

À partir de cet ensemble de cordes  $C$ , le calcul des ouvertures et des granulométries est réalisé d'une manière très efficace :

$$\gamma_L(C) = \{c_j | Length(c_j) \geq L\}, \quad (2.12)$$

$$(PS(C))(L) = \sum_{Length(c_j)=L} V(c_j). \quad (2.13)$$

Grâce à l'équation 2.11,  $\gamma_L(C)$  et  $PS(C)$  sont deux opérateurs strictement équivalents à  $\gamma_L(g)$  et à  $PS(g)$ . On verra par la suite, que cette décomposition du signal en cordes peut être faite d'une manière efficace et rapide, en temps constant par pixel.

L'efficacité de l'algorithme vient de plusieurs caractéristiques : le signal est lu séquentiellement de gauche à droite, chaque corde est analysée une unique fois, et toutes les opérations peuvent être réalisées en  $\mathcal{O}(1)$  par pixel en moyenne y compris la reconstruction du signal 1-D à partir de l'ensemble des cordes (équation 2.11). Le principe de l'algorithme est présenté dans la section suivante.

### 2.3.2 Principe de l'algorithme

En analysant le signal de la figure 2.3, ainsi que son ensemble de cordes  $C$ , on remarque plusieurs phénomènes que l'on décrit ci dessous :

1. Chaque front montant du signal est le point de départ d'au moins une corde.
2. Chaque front descendant du signal représente la fin d'au moins une corde.
3. La longueur d'une corde n'est connue que lorsque l'on arrive à sa fin. Dans la figure 2.3, les portions connues de chaque cordes sont représentées par un trait continu jusqu'à la position de lecture  $rp$ . Les cordes (ou partie de corde) non connues pour l'instant sont en pointillées.
4. Une corde ne pourra être traitée que lorsque la position de lecture  $rp$  arrivera à sa fin.
5. Les cordes naissantes, attendant d'être traitées, peuvent être stockées dans une pile (*LIFO* : *last in, first out*). Une pile est une structure ordonnée qui s'adapte parfaitement à nos données puisque les cordes sont elles-mêmes ordonnées en fonction de leurs tailles et seront donc visitées dans l'ordre d'inclusion.

L'utilisation d'une pile est donc nécessaire pour obtenir un algorithme rapide et la section suivante présente son pseudo code.

### 2.3.3 Pseudo code de l'algorithme

L'algorithme 1 prend pour entrée un signal 1-D ainsi que les paramètres du traitement.  $op$  correspond à l'opérateur que l'on souhaite calculer, il peut prendre comme valeurs  $\{Ouverture, Granulométrie, Arbre\ de\ composantes\ connexes\}$ . Pour le calcul des ouvertures, un paramètre supplémentaire est nécessaire : le paramètre de longueur  $L$ .

En sortie, on aura le résultat de l'opérateur sélectionné : soit  $\gamma_L(g)$ , soit le  $PS$  soit  $CompTree$  pour l'ouverture, la distribution de tailles et l'arbre de composantes connexes respectivement.

$rp$  correspond à la position de la tête de lecture et le signal est lu séquentiellement de gauche à droite, lignes 4 et 5.

---

**Algorithme 1:**  $(g_{out} \text{ or } PS \text{ or } CompTree) \leftarrow ProcessSignal1D(g, L, op)$

---

**Input :**  $g : [1 \dots N] \rightarrow \mathbb{R}$  - signal d'entrée ;

$op$ , opérateur sélectionné

$L$ , paramètre de l'ouverture

**Result :**  $g_{out} = \gamma_L(g)$  or  $CompTree$  or  $PS$

1  $Stack \leftarrow \emptyset$

2  $CompTree \leftarrow \emptyset$

3  $PS[1..N] \leftarrow 0$

4 **for**  $rp = 1..N$  **do**

5    $(g_{out} \text{ or } PS \text{ or } CompTree) \leftarrow ProcessPixel(g(rp), rp, L, Stack, op)$

6  $(g_{out} \text{ or } PS \text{ or } CompTree) \leftarrow ProcessPixel(-\infty, N + 1, L, Stack, op)$

---

Chaque corde est codée par un couple de deux variables  $(sp, k)$  : la position de départ  $sp$  et l'altitude  $k$  de la corde. Les cordes naissantes dont on ne connaît pas encore la taille sont stockées dans une pile appelée  $Stack$  et qui supporte les opérations suivantes :

- $push(sp, k)$  ajout d'une nouvelle corde dans la pile avec une position de départ  $sp$  et une altitude  $k$ ,
- $pop()$  pour supprimer de la pile la dernière corde insérée,
- $empty()$  pour tester si la pile est vide,
- $top().att$  pour lire l'état d'une variable de la dernière corde insérée dans la pile ( $att = sp$  ou  $att = k$ ).

Au début du traitement, la pile est vide et la variable  $PS$  est initialisée à zéro (lignes 1 à 3).

Tous les pixels à la position  $rp$  sont traités par l'algorithme 2 et à chaque nouveau pixel, trois situations peuvent se produire : le pixel introduit un front montant, un front descendant ou une zone plate :

- **Même altitude** : si le pixel courant est à la même altitude que la dernière corde ajoutée dans la pile, alors ce pixel fait partie de cette corde et n'apporte aucune information complémentaire. On passe au pixel suivant.
- **Front montant**, ligne 1 : un front montant signifie que l'on débute une nouvelle corde. On l'ajoute dans la pile en stockant son altitude et sa position de départ, ligne 2.

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

---

---

**Algorithme 2:** ( $g_{out}$  or PS or CompTree)  $\leftarrow$  *ProcessPixel* ( $k$ ,  $L$ ,  $rp$ ,  $Stack$ ,  $op$ )

---

**Input :**  $k = g(rp)$   
     $rp$ , position de la tête de lecture  
     $L$ , paramètre de l'ouverture  
     $op$ , opérateur sélectionné  
     $Stack$ , pile stockant les cordes (LIFO)  
**Result :**  $g_{out}$  or  $PS$  or  $CompTree$  en fonction de  $op$

```
1 if Stack.empty() or k > Stack.top().k then
2   | Stack.push(k,rp,FALSE)
3 else
4   while k < Stack.top().k do
5     | cordOut = Stack.pop()
6     | if op == Granulometrie then
7       | Length = rp - cordOut.sp
8       | PS[Length] += Length × (cordOut.k - max(k,Stack.top().k))
9     | if op == Ouverture then
10      | if cordOut.Passed or rp - cordOut.sp ≥ L then
11        | gout ← WriteCords(cordOut,Stack,rp)
12        | Stack.push(k,rp,TRUE)
13      | break
14    | if op == Arbre de composantes connexes then
15      | if Stack.empty() or Stack.top().k < k then
16        | currentCord.k = k
17        | currentCord.Children.push(cordOut)
18        | Stack.push(currentCord)
19        | break
20      | else
21        | Stack.top().Children.push(cordOut)
22    | if Stack.empty() or k > Stack.top().k then
23      | Stack.push(k,cordOut.sp,FALSE)
24    | break
```

---

---

**Algorithme 3:**  $g_{out} \leftarrow \text{WriteCords}(\text{cordOut}, \text{Stack}, \text{rp})$ 


---

**Input :** *cordOut*, Dernière corde sortie de la pile  
*Stack*, pile stockant les cordes (LIFO)  
*rp*, position de la tête de lecture  
**Result :**  $g_{out} = \gamma_L(g)$

```

1  $g_{out}[\text{cordOut.sp} : \text{rp}] = \text{cordOut.k}$ 
2 while not Stack.empty() do
3    $\text{end} = \text{cordOut.sp}$ 
4    $\text{cordOut} = \text{Stack.pop}()$ 
5    $g_{out}[\text{cordOut.sp} : \text{end}] = \text{cordOut.k}$ 

```

---

- **Front descendant** : un front descendant signifie qu’au moins une corde se termine. Ainsi, la boucle *while* (lignes 4 à 24) dépile toutes les cordes insérées qui viennent de se terminer et les analyse une par une (*cordOut*). À ce niveau, le traitement dépend de l’opérateur sélectionné *op* :
  - **Granulométrie** : pour la corde qui vient d’être dépilée, on calcule sa longueur, ligne 7 puis on ajoute son volume dans l’histogramme indexé par sa longueur, ligne 8. On remarque que si la pile est vide, alors un *front()* sur la pile retourne  $-\infty$ .
  - **Ouverture** : on teste la longueur de la corde avec l’équation 2.12 qui correspond à la ligne 10 de l’algorithme. Dès qu’une corde est plus grande que  $L$ , on reconstruit immédiatement le signal  $g_{out} = \gamma_L(g)$  jusqu’à la position de lecture (avec l’équation 2.11). En effet, la pile est ordonnée grâce à la relation d’inclusion et dès que la longueur d’une corde satisfait le critère de longueur, alors l’ensemble des cordes présent dans la pile satisfait aussi le critère. Il n’est donc pas utile d’attendre leurs fins pour les écrire dans le signal de sortie. D’un point de vue algorithmique, il est même *obligatoire* d’écrire ces cordes dans  $g_{out}$  à ce moment, si on souhaite que chaque pixel ne soit écrit qu’une seule et unique fois dans le signal de sortie. La fonction *WriteCords* fait ce travail : la ligne 1 de l’algorithme 3, écrit la fraction de corde qui vient d’être dépilée dans  $g_{out}$ , puis, la boucle *while* (lignes 2 à 5) va vider la pile. Pour chaque corde, l’écriture se fera au niveau  $k$ , entre le point de départ de cette corde et le point de départ de la corde précédente (ligne 5). L’algorithme 3 vidant la pile et écrivant le signal, il est nécessaire d’ajouter à la structure *corde* un drapeau<sup>1</sup> que l’on nommera *Passed* et qui permettra de se souvenir si la corde courante satisfait le critère de longueur. Enfin, cette corde courante sera placée dans la pile avec une position de départ volontairement erronée (*rp*) afin de ne pas réécrire deux fois la même partie du signal. On voit ici tout l’intérêt de ce drapeau puisque la modification de la position de départ d’une corde entraînera la sous-estimation du calcul de sa longueur. On rappelle que pour des ouvertures, on ne souhaite pas connaître la taille exacte des cordes, mais on souhaite seulement savoir si celles-ci sont plus grande que  $L$ .
  - **Arbre de composantes connexes** : on enrichit la structure *corde* en ajoutant une nouvelle variable appelée *Children* qui correspond à une liste de pointeurs sur la structure *corde*. Le rôle de cette nouvelle variable est de représenter la relation parents/enfants de l’arbre. La corde qui se termine, *cordOut*, est une corde *enfant* qui doit être liée à son parent. Elle peut être liée à deux cordes différentes, la corde courante *currentCord* ou la dernière corde de la pile :

---

1. De l’anglais *flag*. Bit supplémentaire permettant de stocker une information binaire.

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

- Si la pile est vide ou si l’altitude de la dernière corde de la pile est inférieure à l’altitude de la corde courante, alors on lie *cordOut* avec *currentCord* (lignes 16, 17 et 18).
- Sinon, on lie *cordOut* avec la dernière corde de la pile (ligne 21).

Que ce soit pour le calcul des ouvertures, des granulométries ou des arbres de composantes connexes, au moins une corde est présente dans la pile à la fin du signal 1-D. Il est donc nécessaire de traiter ces cordes en fonction de l’opérateur *op* et de la gestion des bords désirée. Nous verrons plus précisément, dans la section 2.4.3, cette gestion des bords. On notera ici que pour construire l’arbre de composantes connexes, il faut dépiler les cordes en les liants à la précédente. La dernière corde restante dans la pile sera appelée la racine<sup>1</sup> de l’arbre. Cette gestion des bords peut être gérée très facilement en ajoutant artificiellement un pixel à  $-\infty$  à la fin du signal (voir la ligne 6 de l’algorithme 1) ce qui permettra de dépiler toutes les cordes restantes dans la pile.

**REMARQUE :** Cet algorithme ne réalise que des opérations de comparaison et de soustraction permettant de l’appliquer sur des données de différentes précisions : 8-bits, 16-bits, 32-bits ou même des flottants simples et doubles précision. Plus généralement en algèbre, cet algorithme peut donc gérer n’importe quel type de données à partir du moment où les valeurs font partie d’un groupe ordonné (pour les trois opérateurs) et muni de la soustraction (seulement pour le calcul des granulométries).

La flexibilité de cet algorithme permet de calculer avec le même pseudo code, des ouvertures avec un attribut de longueur, des granulométries ou l’arbre de composantes connexes. La section suivante présente l’extension de cet algorithme à d’autres opérateurs morphologiques : la reconstruction et l’ouverture par reconstruction. Ces opérateurs nécessitent tous deux, une image marqueur comme entrée, rendant difficile l’intégration des modifications dans le pseudo code 2.

### 2.3.4 Extension à l’opérateur de reconstruction

Choisir des attributs différents permet de définir d’autres opérations morphologiques de base telles que la reconstruction morphologique ou l’ouverture par reconstruction. Avant de présenter les modifications apportées à l’algorithme précédent, nous définissons ces deux opérateurs.

Soit deux signaux mono dimensionnels  $g$  et  $h$  ayant le même support  $D \subseteq \mathbb{Z}$  et le même ensemble de valeurs  $V = \mathbb{Z}$  ou  $\mathbb{R}$  avec  $\forall x \in D, g(x) \geq h(x)$ . La reconstruction morphologique du signal  $h$  sous  $g$  revient en pratique à dilater le signal  $h$ , prendre l’intersection avec le signal  $g$  et à itérer ce processus jusqu’à stabilisation. On note cet opérateur  $Rec_h$  :

$$Rec_h(g) = (\delta h \wedge g)^{(\infty)}. \quad (2.14)$$

La notation  $(.)^{(\infty)}$  signifie que l’on répète le processus jusqu’à idempotence, le symbole  $\wedge$  correspond à l’infimum et le symbole  $\delta$  à une dilatation de taille unitaire.

Avec des cordes maximales, l’équation 2.14 revient à “conserver l’ensemble des cordes de  $g$  tels que, pour une corde  $c$ , il existe au moins un élément de  $h$  dans le support de cette corde, dont le niveau de gris est supérieur ou égale à l’altitude de  $c$ .”

L’ouverture par reconstruction ajoute un critère de taille à la contrainte précédente. La corde  $c$  doit avoir “une longueur supérieure à  $L$  pour survivre et il doit exister au moins un élément de  $h$  dans le support de cette corde, dont le niveau de gris est supérieur ou égale à l’altitude de  $c$ ”.

Ces deux opérateurs sont notés  $Rec_h$  et  $\gamma_{h,L}^{rec}$  et la figure 2.4 illustre leurs résultats sur des signaux  $g$  et  $h$  quelconques.

1. La racine d’un arbre, de l’anglais *root* est le seul élément de l’arbre qui n’a pas d’ancêtre.

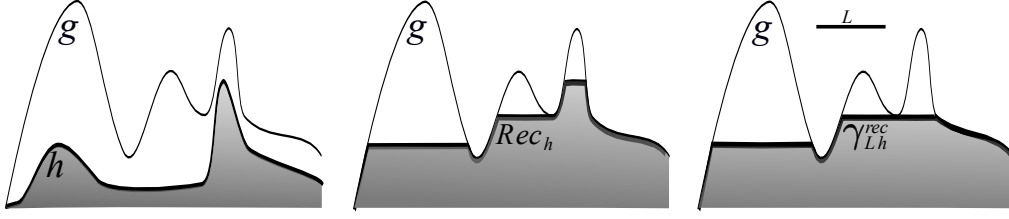


FIGURE 2.4 – Illustration des opérateurs  $Rec_h$  et  $\gamma_{h,L}^{rec}$  sur des signaux 1-D  $g$  et  $h$  quelconques (reconstruction de  $h$  sous  $g$  et l'ouverture par reconstruction)

D'un point de vue algorithmique,  $Rec_h$  est strictement équivalent à  $\gamma_{h,L}^{rec}$  avec  $L = 1$  et le pseudo code 4 présente les modifications nécessaires pour définir  $\gamma_{h,L}^{rec}$ .

La structure *corde* est maintenant composée de 4 attributs : l'altitude de la corde  $k_g$ , l'altitude du marqueur  $k_h$ , la position de départ  $sp$  et le drapeau *Passed*. Lors d'un front montant, on stocke, en plus des attributs usuels, la valeur du marqueur pour ce pixel (lignes 1 et 2). Si le nouveau pixel correspond à la même corde, alors on met à jour la valeur du marqueur en prenant la valeur maximale (lignes 3 et 4). Enfin, pour un front descendant, les cordes se terminant sont dépilées et traitées successivement (lignes 6 à 16). On teste donc si la longueur de la corde est supérieure à  $L$  et si la valeur de son marqueur est supérieure ou égale à l'altitude de la corde père. Si oui, il faut écrire cette corde à une altitude égale à la valeur du marqueur. De la même manière que pour les ouvertures par attributs de longueur, si une corde satisfait le critère, alors il en est de même pour toutes les cordes présentes dans la pile et elles sont écrites dans le signal de sortie avec la fonction *WriteCordReconstruction* (algorithme 5). Enfin, si la corde ne satisfait pas le critère, la valeur du marqueur est mise à jour avec des opérations de min et de max afin de s'assurer que cette valeur sera toujours la plus haute possible tout en restant inférieure au signal original  $g$  (lignes 10 à 16).

## 2.4 Extension aux signaux 2-D

Cette section présente l'extension des algorithmes 1-D précédemment développés, à des images 2-D. Nous verrons comment le support de l'image sera partitionné en lignes droites mono dimensionnelles, suivant une orientation quelconque. Nous utiliserons la même stratégies pour calculer des granulométries orientées [131]. Enfin, nous aborderons la gestion des bords de l'image puis la construction de l'arbre de composantes connexes 2-D.

Dans cette section,  $f : D \rightarrow V$  est une image 2-D avec  $D \subseteq \mathbb{Z}^2$ , un support rectangulaire.

### 2.4.1 Stratégies de balayage pour des images 2-D

Prenant en entrée un pixel, l'algorithme 2 est indépendant de l'orientation de la ligne. Il peut donc être étendu sans difficulté à des images 2-D, en utilisant un balayage adapté. Formellement, dans la théorie de la géométrie quasi affine [165], l'image  $f$  est décomposée en  $k$  signaux 1-D,  $\{f_{\alpha,k}\}_{k \in K}$ , suivant la direction  $\alpha$ . Soille *et al.* [213] ont défini un parcours astucieux de l'image avec des lignes de Bresenham [30], qui a pour but de supprimer tous les chevauchements de pixels entre deux lignes successives. Ainsi, chaque pixel de l'image n'est utilisé qu'une unique fois, laissant la possibilité d'avoir un traitement *in place*. L'extraction des lignes avec une orientation arbitraire demande du temps de

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

---



---

**Algorithme 4:**  $g_{out}^{rec} \leftarrow ProcessPixelReconstruction(k_g, k_h, L, rp, Stack)$

---

**Input :**  $k_g = g(rp)$   
 $k_h = h(rp)$   
 $rp$ , position de la tête de lecture  
 $L$ , paramètre de l'ouverture  
 $Stack$ , pile stockant les cordes (LIFO)  
**Result :**  $g_{out}^{rec}$

```

1 if Stack.empty() or  $k_g > Stack.top().k_g$  then
2   |  $Stack.push(k_g, k_h, rp, FALSE)$ 
3 else if  $k_g == Stack.top().k_g$  then
4   |  $Stack.top().k_h = \max(Stack.top().k_h, k_h)$ 
5 else
6   |  $cordOut = Stack.pop()$ 
7   | if  $cordOut.Passed$  or
8     |  $(rp - cordOut.sp \geq L$  and  $cordOut.front().k_h \geq \max(Stack().front().k_g, k_g))$  then
9     |    $g_{out}^{rec} \leftarrow WriteCordsReconstruction(cordOut, Stack, rp, cordOut.top().k_h)$ 
10    |    $Stack.push(k_g, k_h, rp, TRUE)$ 
11 else if  $Stack.empty()$  or  $k_g > Stack.top().k_g$  then
12   |  $Stack.push(k, \min(k_g, \max(k_h, cordOut.top().k_h)), cordOut.sp, FALSE)$ 
13 else if  $k_g == Stack.top().k_g$  then
14   |  $Stack.top().k_h = \min(Stack.top().k_g, \max(cordOut.top().k_h, Stack.top().k_h))$ 
15 else
16   |  $Stack.top().k_h = \min(Stack.top().k_g, \max(cordOut.top().k_h, Stack.top().k_h))$ 
17   | Goto Line 6

```

---



---

**Algorithme 5:**  $g_{out}^{rec} \leftarrow WriteCordsReconstruction(cordOut, Stack, rp)$

---

**Input :**  $cordOut$ , dernière corde sortie de la pile  
 $Stack$ , pile stockant les cordes (LIFO)  
 $rp$ , position de la tête de lecture  
 $k_h$ , niveau maximal du marqueur  
**Result :**  $g_{out}^{rec} = \gamma_L(g)$

```

1  $g_{out}^{rec}[cordOut.sp : rp] = \min(cordOut.top().k_g, k_h)$ 
2 while not Stack.empty() do
3   |  $end = cordOut.sp$ 
4   |  $cordOut = Stack.pop()$ 
5   |  $g_{out}^{rec}[cordOut.sp : end] = \min(cordOut.top().k_g, k_h)$ 

```

---

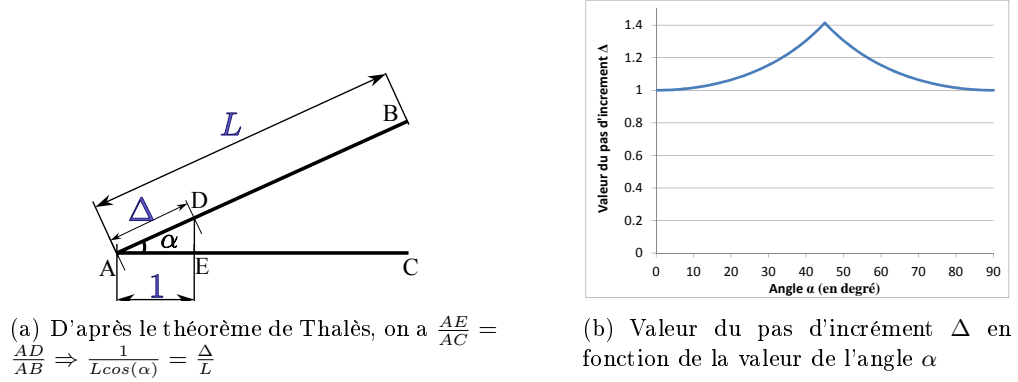


FIGURE 2.5 – Stratégie mise en place pour prendre en compte la discrétisation du support 2-D.

calcul. On ajoute donc une ligne mémoire stockant les coordonnées de chaque pixel appartenant au signal 1-D. Ce buffer servira uniquement pour les ouvertures, pour écrire dans l'image de sortie sans avoir à recalculer la position des différents pixels de la ligne.

Sur une trame discrète, la précision de cet opérateur est détériorée si on ne prend pas en compte l'orientation de la ligne. En effet, pour des lignes orientées suivant  $45^\circ$ , la distance parcourue entre chaque pixel est égale à  $\sqrt{2}$  contre 1 pour des lignes horizontales. Pour améliorer la précision de cet opérateur, Hendriks propose de “tordre” l'image [127] afin de toujours calculer des ouvertures horizontales tandis que Soille et Talbot [215] proposent des ouvertures 1-D invariantes par translation. Nous proposons de rendre ces opérateurs invariants par rotation en utilisant une méthode un peu différente. La figure 2.5(a) présente un segment de longueur  $L$  et d'orientation  $\alpha$  et nous cherchons à déterminer le pas d'incrément optimal noté  $\Delta$ . D'après le théorème de Thalès, on obtient :

$$\Delta(\alpha) = \begin{cases} \frac{1}{\cos(\alpha)}, & 0^\circ \leq \alpha < 45^\circ, \left[\frac{\pi}{2}\right] \\ \frac{1}{\sin(\alpha)}, & 45^\circ \leq \alpha < 90^\circ, \left[\frac{\pi}{2}\right], \end{cases} \quad (2.15)$$

si la résolution spatiale est la même en  $x$  et en  $y$ . La figure 2.5(b) propose de tracer  $\Delta$  en fonction de l'angle  $\alpha$ . Utiliser  $\Delta(\alpha)$  comme pas d'incrément pour chaque pixel permet de correctement mesurer la longueur des chemins diagonaux. Pour montrer l'intérêt de cette méthode, on propose une expérience dans la section 3.6.1 (figure 3.12(a)) qui confirme la propriété d'invariance par rotation de cet opérateur.

### 2.4.2 Granulométries orientées

Nous avons vu précédemment que l'orientation locale d'un pixel d'une image 2-D peut être mesurée grâce au supremum des ouvertures par segments. Pour compléter cette mesure, on peut calculer des granulométries pour chaque orientation  $\alpha$  des lignes, permettant d'obtenir un opérateur introduit par Maragos [131] : *Oriented Pattern Spectrum* (*OPS*) :

$$OPS(f)(\lambda, \alpha) = \sum_{k \in K} (PS(f_{\alpha, k}))(\lambda). \quad (2.16)$$

Le calcul de l'*OPS* peut être très long et utiliser l'algorithme rapide présenté dans ce chapitre permet de réduire le temps de calcul. Pour avoir un outil visuel, le résultat de l'*OPS* est un histogramme 2-D,



## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

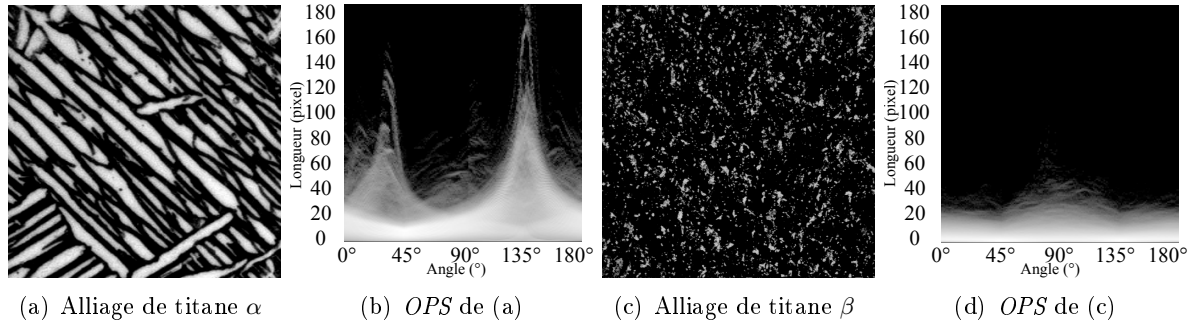


FIGURE 2.6 – Granulométries orientées pour des alliages de titane  $\alpha$  et  $\beta$ . Voir le texte pour les explications.

qui sera présenté comme une image 2-D où chaque colonne correspond à une orientation  $\alpha$  et où le niveau de gris correspond au logarithme du volume cumulé, normalisé entre 0 et 255. La figure 2.6 présente deux images d'alliage de titane, type  $\alpha$  et  $\beta$  (Fig. 2.6(a) et 2.6(c)) et les résultats de leurs *OPS* respectifs (Fig. 2.6(b) et 2.6(d)). Dans l'*OPS* de la figure (b), on remarque la présence de deux pics montrant que les structures sont orientées suivant deux axes principaux à  $140^\circ$  et  $40^\circ$ . La taille des structures, largeur et longueur, peut aussi être déduite de cet histogramme 2-D : jusqu'à 180 pixels de longueur et 15 pixels de largeur pour les fibres orientées à  $140^\circ$ . Pour le titane  $\beta$ , l'orientation est plutôt isotrope avec cependant, un léger alignement pour les orientations à  $90^\circ$ .

### 2.4.3 Effets de bord

Pour discuter de l'effet de bord, on considère un processus aléatoire et stationnaire,  $\xi$ , de distribution spatiale uniforme de segments de droites d'une longueur  $l$  pixels, orientés dans une direction  $\vartheta$ . Le placement est aléatoire dans l'espace cependant, on ajoute une contrainte aux segments de droites qui ne doivent pas se chevaucher. Ainsi, le pattern spectrum  $PS(\xi)$  dans la direction  $\vartheta$  est  $d(l)$ , la distribution de Dirac à une longueur  $l$ . On considère maintenant le même processus  $\xi$  avec un domaine de l'image fini  $D \subset \mathbb{Z}^2$ , figure 2.7. L'intersection du processus  $\xi$  avec un support fini introduit un biais dans le PS (effet de bord) puisque quelques structures apparaissent coupées dans  $D$  et contribuent pour des longueurs inférieures (voir la courbe de la figure 2.7).

L'effet de bords est un problème récurrent en traitement de l'image qui est géré différemment suivant les applications. Van Droogenbroeck [239] propose une discussion intéressante sur ce problème et recommande d'ajouter la valeur maximale de  $V$  à l'extérieur du support de l'image. On note cette valeur  $+\infty$  par commodité. Cependant, dans le domaine des filtres connexes, la valeur minimale de  $V$  est souvent retenue comme extension du domaine  $D$  (notée  $-\infty$ ). On détaille ici les conséquences de ces deux extensions :

- Ajouter des valeurs  $+\infty$  en dehors du support revient à considérer que les structures touchant le bord de l'image ont une longueur infinie. Ainsi, elles résisteront à toutes les ouvertures avec  $L < \infty$ . Pour les granulométries, du fait que ces structures ne sont jamais supprimées, elles n'interviendront pas dans la distribution de taille (courbe bleue de la figure 2.7). Cela est justifiée puisque l'on ne connaît pas la longueur des structures touchant le bord de l'image. Néanmoins,

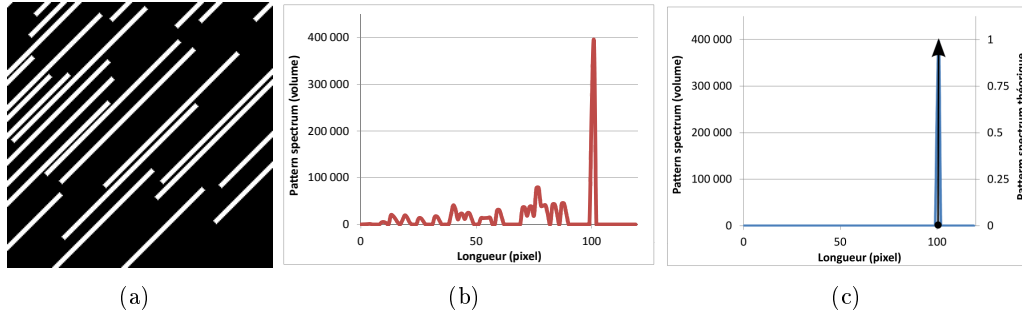


FIGURE 2.7 – Illustration de l’effet de bord. (a) on place aléatoirement des lignes droites de longueur 100 pixels avec une orientation de  $45^\circ$  puis on calcule l’intersection avec un support fini. (b) la courbe en rouge permet de visualiser l’effet de bords et le biais introduit sur le PS. (c) le pattern spectrum théorique est un dirac centré sur 100 et le PS expérimental utilisant l’extension  $+\infty$  (en bleu).

un biais est introduit puisque les grandes structures ont une probabilité plus grande de toucher un bord.

- Ajouter des valeurs  $-\infty$  en dehors du support revient à considérer que les structures que l’on observe sont complètes (même celles qui sont tronquées par un des bords de l’image). Aussi, elles peuvent être supprimées avec des ouvertures et elles contribueront dans la distribution de taille (courbe rouge dans la figure 2.7(b)). Cette extension du support introduit aussi un biais puisque les structures touchant le bord de l’image ont une longueur plus longue que celle retenue.

Aucune de ces deux méthodes ne permet de s’affranchir des biais dans les mesures, aussi, l’algorithme de ce chapitre propose de gérer ces deux extensions du support. Pour l’extension  $-\infty$ , on utilise le pseudo code 2. Pour l’extension  $+\infty$ , des légères modifications sont nécessaires. Pour les ouvertures, le drapeau *Passed* est mis à vrai pour toutes les cordes touchant le support du signal. Pour des granulométries, il faut supprimer la ligne 6 de l’algorithme 1 et faire un test supplémentaire sur la position de départ de la corde, pour savoir si celle-ci contribue ou non au *PS*.

En stéréologie, d’autres méthodes sont utilisées et on redirige le lecteur vers [159]. On note néanmoins que la correction de Miles-Lantuéjoul [146, 114] permet d’obtenir une mesure non biaisée en supprimant toutes les particules incomplètes, touchant le bord de l’image, (extension  $+\infty$ ) et en pondérant chaque particule par un poids proportionnel à la probabilité que cette dernière soit entièrement incluse dans le champ observé.

## 2.5 Complexité

Nous étudions dans cette section la complexité de cet algorithme en focalisant notre attention tout d’abord sur la complexité du signal 1-D, puis sur celle de l’algorithme 2-D.

### 2.5.1 Signal 1-D

On considère le même signal 1-D de taille  $N$  que dans la section 2.3,  $g : [1...N] \rightarrow V$ , avec  $V = \mathbb{R}$  ou  $\mathbb{Z}$ . On rappelle que le signal est lu séquentiellement de la gauche vers la droite (lignes 4 et 5 de l’algorithme 1) et que l’algorithme 2 est appelé une fois par échantillon du signal.

En analysant l’algorithme 2, on remarque que chaque corde est empilée une unique fois lorsqu’elle débute (ligne 2), puis dépilée lorsque celle-ci se termine (ligne 5). Lors d’un front descendant, plusieurs

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

---

cordes peuvent se terminer en même temps. Elles seront néanmoins traitées séquentiellement dans la boucle *while*. On note que cette boucle ne sera exécutée qu’une fois par corde. Comme toutes les autres opérations de l’algorithme 2 (tests, arithmétique, opérations sur la pile) sont calculées en temps constant, une corde est donc traitée en  $\mathcal{O}(1)$  par pixel en moyenne. On en déduit aussi que la complexité dépend du nombre de cordes de l’image.

En ce qui concerne la consommation mémoire, une corde est stockée sur 3 à 10 octets (pour des lignes ayant au maximum 65536 pixels et pour des données d’entrée de 1 à 8 octets) et le cas le plus défavorable est un signal strictement croissant. Dans ce cas précis, on insérera dans la pile  $N$  cordes. Néanmoins, la quantité de mémoire requise par cet algorithme reste très faible et peut être bornée plus précisément. Au maximum, le nombre d’éléments que l’on peut insérer dans la pile correspond au minimum entre la taille du signal et le nombre de niveaux de gris :  $\min(N, \text{card}(V))$ .

### 2.5.2 Signal 2-D

On effectue un balayage complet de l’image rectangulaire 2-D avec un ensemble de lignes de Bresenham, parallèles et orientées suivant une direction  $\alpha$ . Avec l’algorithme proposé par Soille [213], l’extraction des lignes 1-D est réalisée en  $\mathcal{O}(1)$  par pixel.

On note que chaque ligne est indépendante des autres, ce qui permet d’introduire du parallélisme dans ces opérateurs. La section 2.7 décrit le fonctionnement de tels algorithmes.

Ainsi la complexité n’augmente pas en prenant une image 2-D et par conséquent, une ouverture, une granulométrie ou un arbre de composantes connexes auront une complexité constante par pixel, ou linéaire par rapport au nombre de pixels de l’image. La section suivante propose de vérifier expérimentalement la complexité de cet algorithme et regroupe l’ensemble des benchmarks réalisés.

## 2.6 Temps d’exécution et comparaison avec l’état de l’art

Outre la grande flexibilité de cet algorithme, sa vitesse d’exécution est un facteur déterminant. On propose donc dans cette section, de quantifier sa vitesse en effectuant des benchmarks complets avec les algorithmes de référence de l’état de l’art.

Tous les tests de cette section ont été réalisés en utilisant uniquement un *thread* d’un ordinateur portable : Intel Core 2 Duo T7700 CPU @2.4GHz. La parallélisation de cet algorithme est expliquée dans la section 2.7.

### 2.6.1 Temps d’exécution pour les ouvertures

On propose 6 benchmarks pour tester la vitesse des ouvertures sur des images 2-D. Tout d’abord, nous verrons l’influence du contenu de l’image sur les temps de calcul en proposant deux tests : le premier utilisera des images quelconques, et le second des images bruitées puis simplifiées par un filtre gaussien. Enfin, nous situerons notre algorithme par rapport à d’autres implémentations rapides des ouvertures en testant plusieurs facteurs, la taille  $L$ , l’orientation, la taille de l’image et le type de données d’entrée.

#### Benchmarks avec le contenu de l’image

Après avoir étudié la complexité de l’algorithme dans la section 2.5, nous avons vu que les temps d’exécution dépendent du contenu de l’image. Pour s’en assurer, nous mesurons les temps moyens pour

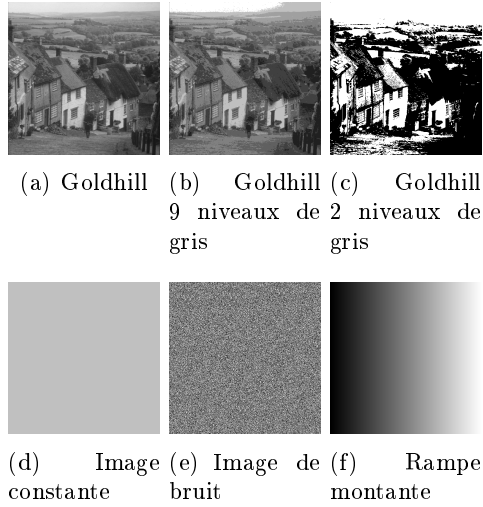


FIGURE 2.8 – Images utilisées pour le calcul du graphique 2.9

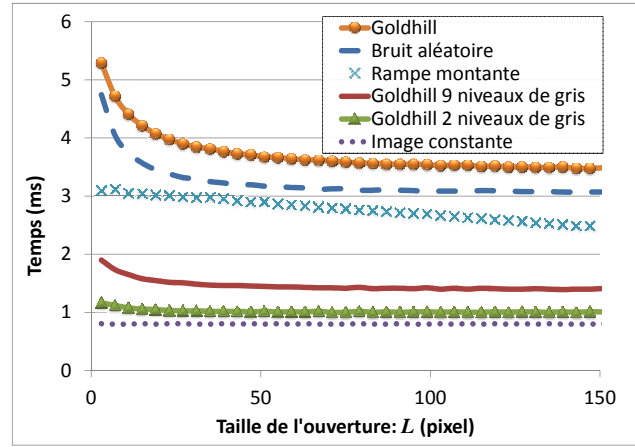


FIGURE 2.9 – Temps de calcul pour une ouverture horizontale de taille  $L$  pour différentes images 512 par 512 pixels

calculer 1000 ouvertures horizontales en fonction de la taille des ouvertures  $L$  et sur différentes images 512 par 512 pixels : une image naturelle Goldhill, deux autres versions de Goldhill (où le nombre de niveaux de gris a été réduit à 2 et à 9), une image constante, une image de bruit uniformément distribué et une image d'une rampe en niveaux de gris. Ces images sont regroupées dans la figure 2.8.

La figure 2.9 regroupe l'ensemble des résultats pour le test décrit précédemment. On remarque tout d'abord que les temps de calcul ne sont pas complètement indépendants de  $L$ . En effet, ils sont réduits lorsque  $L$  augmente ce qui peut paraître contre intuitif à première vue. Cependant, l'explication vient du fait que des ouvertures de grandes tailles suppriment un nombre important de cordes dans le signal ce qui permet d'avoir des écritures dans l'image de sortie qui sont moins fréquentes. Un nombre réduit de phases d'écritures a pour conséquence d'accélérer l'algorithme. Ceci est confirmé par le signal de la rampe montante. Dans cette image, de nombreuses cordes sont empilées dans la pile, mais il y a une unique phase d'écriture expliquant les temps de calcul relativement faibles par rapport aux autres images.

Ce test prouve clairement que la complexité de l'algorithme dépend du contenu de l'image. Comme attendu, une image constante définit la borne inférieure de cet algorithme. De plus, il semble que plus il y ait de cordes dans une image (et de phases d'écriture) et plus les temps de calcul sont élevés. Pour s'en convaincre, nous proposons un test permettant de tracer le temps d'exécution d'une ouverture horizontale en fonction du nombre ou de leurs longueurs moyennes des cordes de l'image. Pour cela, une image bruitée (bruit blanc uniforme sur l'intervalle  $[0,255]$ ) va être simplifiée grâce à un filtre gaussien pour différentes valeurs d'écart type de  $(0, 1, \dots, 19)$ . Plus le filtrage sera important et plus le nombre de cordes diminuera et plus leur longueur moyenne sera grande.

Quelques images utilisées sont regroupées dans la figure 2.10 et les résultats sont présentés sous forme de deux graphiques (figures 2.11(a) et 2.11(b)). On remarque que les courbes semblent fortement corrélées. Néanmoins, pour connaître le paramètre qui a la plus grande influence sur le temps d'exécution des ouvertures, on calcule le coefficient de corrélation entre la longueur moyenne des cordes et les temps de calcul (0,965), puis entre le nombre de cordes et les temps de calcul (−0,933). Même

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

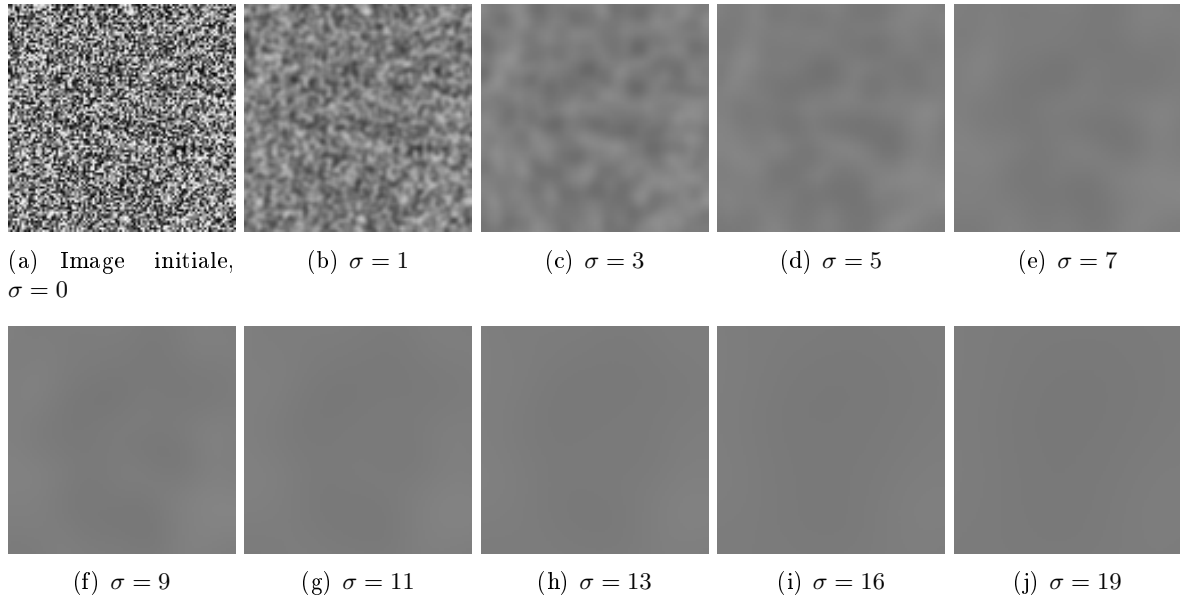


FIGURE 2.10 – Exemple d’images 512 par 512 utilisées pour générer les graphiques de la figure 2.11 (Pour faciliter l’affichage, leurs tailles sont réduites à 100 par 100 pixels). Suivant la valeur de l’écart type du filtre gaussien, l’image est plus ou moins lissée.

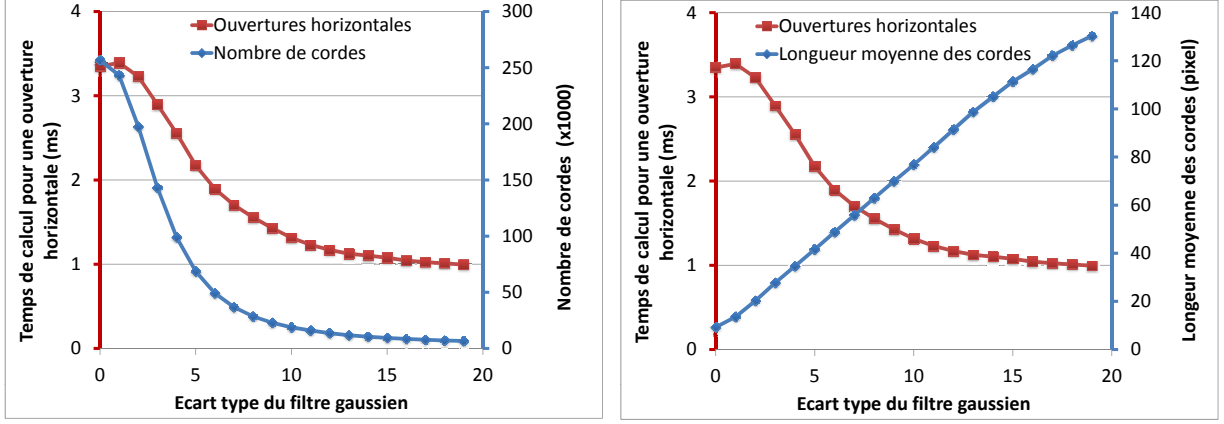
si ces courbes sont très corrélées, c’est toutefois le nombre de cordes qui influence le plus les temps d’exécution de cet algorithme, ce qui corrobore les conclusions tirées de l’analyse de la figure 2.9 : **moins il y a de cordes, moins il y a d’interruptions pour écrire le résultat de l’ouverture et plus les temps d’exécution sont rapides.**

### Benchmark avec d’autres algorithmes

Pour situer notre algorithme et le comparer avec l’état de l’art, nous utilisons 5 autres algorithmes pour calculer des ouvertures 1-D :

1. L’algorithme en  $\mathcal{O}(L)$  suivant la définition de Serra [202] appelé *Naïf* dans la suite de ce chapitre (Calcul de l’érosion suivie de la dilatation).
2. L’algorithme en  $\mathcal{O}(1)$  de Van Herk[241], Gil et Werman [77] appelé *HGW* (Calcul de l’érosion suivie de la dilatation).
3. L’algorithme en  $\mathcal{O}(1)$  de Clienti *et al.* [39] appelé *Clienti* (Calcul de l’érosion suivie de la dilatation).
4. L’algorithme en  $\mathcal{O}(1)$  de Bartovsky *et al.* [13] appelé *Bartovsky* (Calcul de l’ouverture directement).
5. L’algorithme en  $\mathcal{O}(1)$  de Van Droogenbroeck et Buckley [239, 46] appelé *Van Droogenbroeck* (Calcul de l’ouverture directement).

Tous ces algorithmes de ce benchmark ont été, soit fournis par l’auteur de l’algorithme, soit réimplémentés tout en restant fidèle au papier original. Ils sont écrits en C/C++ et intégrés dans la même



(a) Evolution du temps de calcul et du nombre de cordes en fonction l'écart type du filtre gaussien (b) Evolution du temps de calcul et de la longueur moyenne des cordes en fonction l'écart type du filtre gaussien

FIGURE 2.11 – Evolution du temps de calcul moyen pour des ouvertures horizontales de longueur  $L = \{1, \dots, 100\}$  pixels en fonction du contenu de l'image.

plateforme de traitement d'images avec exactement la même interface. Pour chaque algorithme, on mesure les temps d'exécution moyens pour calculer 1000 ouvertures avec des orientations quelconques pour différents  $L$ . Les résultats reportés sur la figure 2.12 ont été calculés sur l'image Goldhill 8-bits, mais on note que les conclusions sont les mêmes avec d'autres images 8-bits.

On constate que suivre à la lettre la définition d'une ouverture ne permet pas d'obtenir des performances satisfaisantes. En effet, la complexité n'est pas la même entre l'implémentation naïve et les autres algorithmes et par conséquent, les écarts deviennent vite abyssaux lorsque l'on augmente la taille de l'ouverture.

On constate aussi que la vitesse d'exécution de notre algorithme est très bonne cependant, l'algorithme de Van Droogenbroeck reste le plus rapide de tous, lorsque la taille de l'ouverture est supérieure à 10 pixels. On peut avancer une raison principale pour expliquer cette vitesse. Avant d'analyser la ligne 1-D, le programme recopie cette ligne dans l'image de sortie en utilisant la fonction *C memcpy*, puis travaille *in place*. Enfin, il n'écrit que les pixels qui doivent être modifiés. On a vu précédemment que ce sont les phases d'écritures qui ralentissent l'algorithme, et ceci explique les différences de temps de calcul.

On note cependant, que l'algorithme de Van Droogenbroeck est basé sur un histogramme. Or, cette structure est inefficace pour des données d'entrée ayant une précision supérieure à 8 bits. Il sera alors préférable d'utiliser une file de priorité (*priority queue*) pour éviter de scanner entièrement un très gros histogramme. Par conséquent, cet algorithme sera beaucoup moins rapide pour des données 16-bits, 32-bits ou des réels.

### Benchmark avec l'orientation des lignes 1-D

Les ouvertures linéaires avec des orientations quelconques nécessitent d'extraire un ensemble de lignes des images 2-D. Nous verrons dans ce paragraphe quelle est l'influence de l'orientation des lignes dans le calcul des ouvertures directionnelles. Nous utilisons la même méthode pour extraire les lignes pour tous les algorithmes, exceptés l'algorithme Bartovsky, qui utilise une méthode propre. On mesure

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

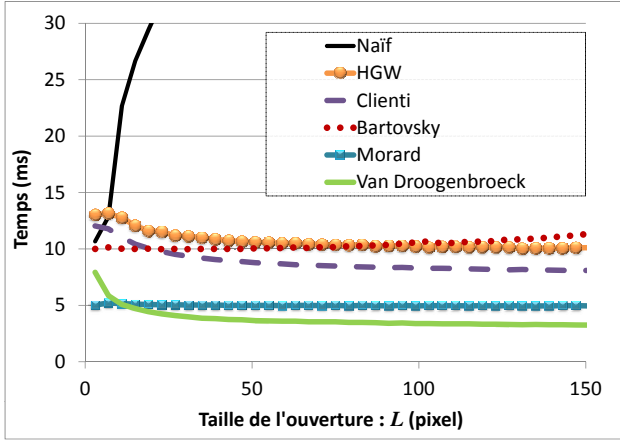


FIGURE 2.12 – Temps de calcul moyen pour une ouverture avec une orientation quelconque en fonction de la taille  $L$ , pour différents algorithmes. L'image utilisée est Goldhill (8-bits, 512 par 512 pixels).

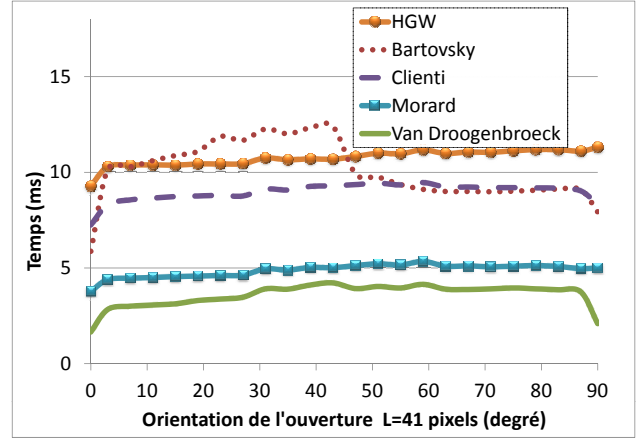


FIGURE 2.13 – Temps de calcul moyen pour une ouverture de taille  $L = 41$  pixels en fonction de l'orientation des lignes, pour différents algorithmes. L'image utilisée est Goldhill (8-bits, 512 par 512 pixels).

le temps pour calculer 1000 ouvertures que l'on moyenne pour tous les angles  $\alpha = [0^\circ, 90^\circ]$ . La taille de l'ouverture  $L$  est fixée à 41 pixels et on utilisera l'image Goldhill 8-bits.

La figure 2.13 regroupe les courbes pour les différents algorithmes considérés et on remarque que les temps de calcul sont presque indépendants de l'orientation des lignes. Cependant, pour  $\alpha = 0^\circ$  ou  $\alpha = 90^\circ$ , les ouvertures sont plus rapides puisque l'extraction des lignes ne nécessite aucun calcul. La configuration la plus favorable pour l'ensemble des algorithmes considérés, est une orientation de  $0^\circ$ . Ceci est dû à l'organisation des données en mémoire. Chaque ligne est stockée l'une après l'autre et deux pixels voisins d'une ligne horizontale sont contigus en mémoire. Une ouverture horizontale minimise donc les *défauts de caches* (*cache miss*).

### Benchmark avec la taille de l'image

Pour voir l'influence de la taille de l'image sur les temps de calcul des ouvertures, on réalise un autre test. On choisit l'image Goldhill, que l'on réplique afin d'obtenir des images de différentes tailles (512x512, 1024x1024, 2048x2048 et 4096x4096) mais avec les mêmes structures dans l'image. Enfin, nous mesurons le nombre de pixels moyen que nous pouvons traiter en une seconde (1000 réalisations) pour des ouvertures horizontales de tailles comprises entre 1 et 200 pixels. L'unité de cette mesure est en  $\text{Pixels.s}^{-1}$ .

On cherche à visualiser l'évolution de la performance des ouvertures lorsque la taille de l'image augmente. Ainsi, la figure 2.14 regroupe les évolutions relatives du nombre de pixels par seconde traités par les différents algorithmes. L'image référence est l'image Goldhill de base (512x512). On constate que l'algorithme proposé évolue positivement lorsque l'on augmente la taille de l'image. Plus l'image est grande et plus on pourra traiter de pixels en une seconde. De tous les algorithmes testés, cette évolution des performances est la meilleure.

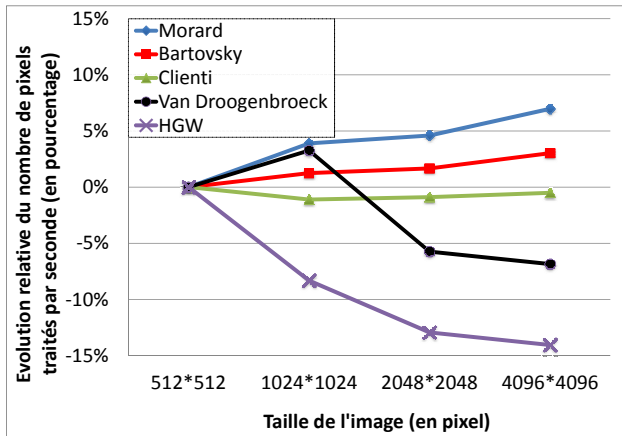


FIGURE 2.14 – Evolution relative du nombre de pixels traités par seconde en fonction de la taille de l'image par rapport à une image référence Goldhill 512 x 512.

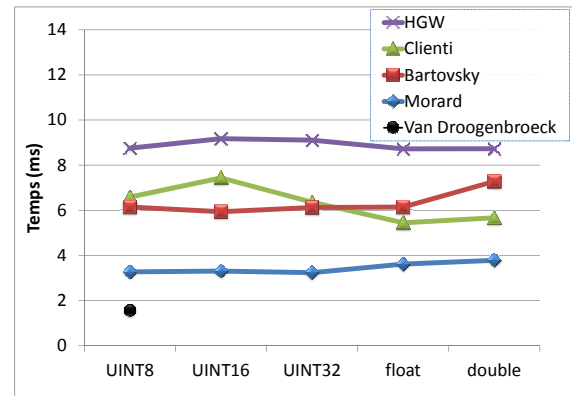


FIGURE 2.15 – Temps de calcul pour des ouvertures arbitraires en fonction du type de données d'entrée (Image 512 x 512 pixels de bruit uniforme).

## Benchmark avec la précision des données d'entrée

Tous les tests précédents ont été réalisés avec des images 8-bits. Ce benchmark propose de visualiser l'influence de la précision du type d'entrée sur les temps de calcul. On calculera des ouvertures horizontales sur des images 8-bits, 16-bits, 32-bits et des flottants (simples et doubles précisions). Pour s'affranchir du contenu de l'image, nous utilisons une image de bruit uniformément distribué de taille 512x512 pixels. Les temps de calcul de 1000 ouvertures sont moyennés pour chaque type d'entrée et les résultats sont présentés dans la figure 2.15. On note que l'algorithme de Van Droogenbroeck, bien que plus rapide pour des images 8-bits, ne supporte pas d'autres types d'entrée. On remarque que l'algorithme proposé est le plus rapide pour des ouvertures 1-D avec des données d'entrée ayant une précision supérieure à 8-bits.

### 2.6.2 Temps d'exécution pour les granulométries

Comparé à une implémentation naïve, l'algorithme de Vincent pour des granulométries réduit les temps de calcul par plusieurs ordres de grandeur [246]. Cet algorithme a été introduit en 2000 et il est toujours le plus rapide pour le calcul des granulométries 1-D. Dans ce test, nous avons comparé trois algorithmes (celui de Vincent, Bartovsky et le notre) en mesurant le temps moyen pour calculer 1000 granulométries horizontales pour des images de bruit de tailles différentes. Les résultats sont présentés dans la figure 2.16 et on observe que notre algorithme est le plus rapide. Les temps de calcul ont été réduit de plus de 26% par rapport à l'algorithme de Vincent et de plus de 245% par rapport à l'algorithme de Bartovsky.

Pour information, les temps de calcul pour le pattern spectrum orienté des images 2.6(a) (322x322) et 2.6(c) (625x625) sont respectivement de 0,37 et 1,06 secondes pour 180 orientations.



## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

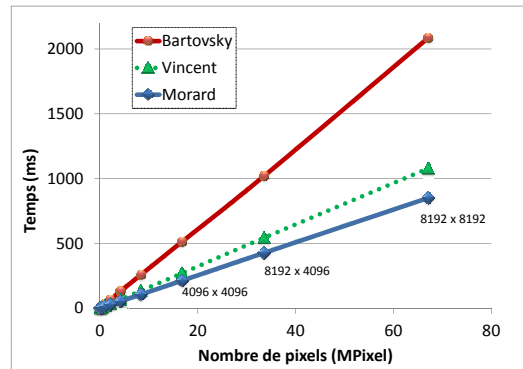


FIGURE 2.16 – Temps de calcul pour des granulométries horizontales en fonction de la taille de l'image. L'algorithme proposé est plus rapide que celui de Vincent ou celui de Bartovsky (Image de bruit uniforme).

## 2.7 Parallélisation de l'algorithme

L'algorithme présenté dans ce chapitre travaille exclusivement sur des signaux 1-D. Par conséquent, chaque ligne est indépendante des autres et on peut facilement envisager un calcul parallèle. Cette section montre les méthodes utilisées pour paralléliser cet algorithme sur (i) l'unité centrale de traitement (CPU) puis sur (ii), la carte graphique (GPU).

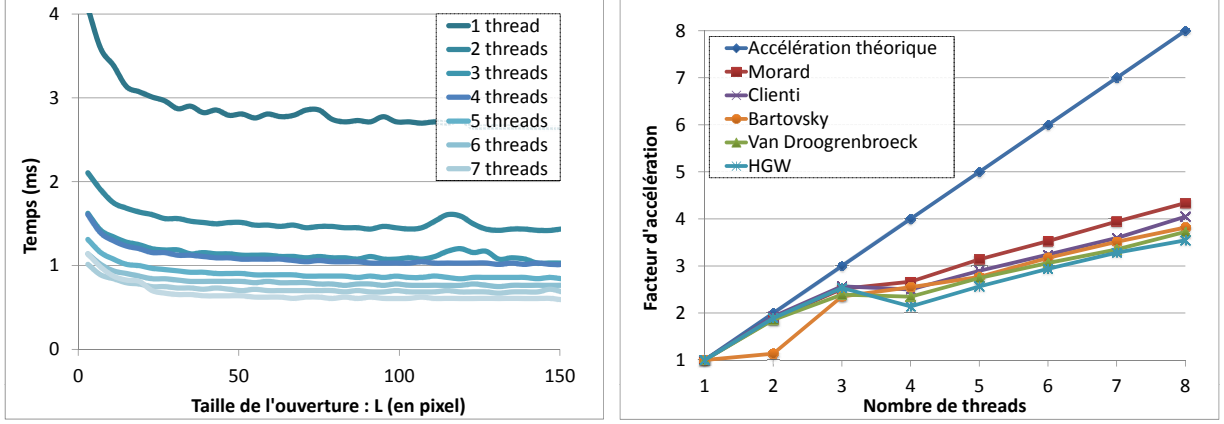
### 2.7.1 Parallélisation sur CPU

Les ordinateurs récents possèdent plusieurs cœurs qui peuvent à leur tour, exécuter plusieurs threads<sup>1</sup>. Un processeur intel core i7 2860-QM CPU @2.50GHz possède 4 cœurs pouvant chacun exécuter 2 threads. Ainsi, nous utiliserons un ordinateur portable muni de ce processeur pour réaliser des benchmarks sur les gains de la parallélisation, puisqu'il peut exécuter simultanément jusqu'à 8 threads.

On restreint l'orientation des lignes en imposant  $\alpha = 0^\circ$  et  $\alpha = 90^\circ$ . Pour obtenir un opérateur parallèle efficace, on divise le domaine de l'image en bandes de tailles égales, horizontales ou verticales en fonction de l'angle  $\alpha$ , puis chaque thread calculera indépendamment le résultat des ouvertures, des granulométries ou des arbres de composantes connexes sur ce sous domaine. La figure 2.17(a) présente les temps de calcul moyen d'une ouverture horizontale en fonction de la taille  $L$  et en fonction du nombre de threads utilisés (de 1 à 8). L'image Goldhill 512x512 8-bits sera l'image test et on note, comme attendu, que les temps de calcul diminuent avec l'augmentation de la parallélisation de l'algorithme.

On applique le même processus à l'ensemble des algorithmes permettant de calculer les ouvertures 1-D et on trace l'évolution du facteur d'accélération en fonction du nombre de threads utilisé (figure 2.17(b)). On constate tout d'abord que l'accélération obtenue n'est pas linéaire par rapport au nombre de threads mais on note que l'augmentation maximale de 433% est obtenue avec notre algorithme.

1. Un thread est une unité de traitement d'un processus, ou processus léger. Utiliser des threads fournit un parallélisme fin et permet de détailler l'exécution d'un processus lourd.



(a) Temps de calcul moyen en fonction de  $L$  et du nombre de threads (b) Facteur d'accélération en fonction du nombre de threads pour différents algorithmes

FIGURE 2.17 – Parallélisation de l'algorithme sur le CPU pour une ouverture horizontale.

La parallélisation de cet algorithme avec des orientations arbitraires est plus difficile mais possible puisque l'algorithme d'extraction des lignes de Soille *et al.* [213] assure que chaque pixel ne sera traité qu'une seule fois.

Cependant, de meilleures performances peuvent être obtenues si l'on déporte le calcul des ouvertures vers la carte graphique.

### 2.7.2 Parallélisation sur GPU

Utiliser une carte graphique pour des ouvertures ou des granulométries permet de bénéficier de sa capacité de traitement massivement parallèle pour réduire les temps de calcul. Initialement, les cartes graphiques étaient uniquement conçues pour assurer les fonctions de calcul liées à l'affichage. Puis les cartes graphiques devinrent programmable et commencèrent à être utilisées à d'autres fins (GPGPU)<sup>1</sup>, vers les années 2000. Des plateformes de développement en C, tels que CUDA par nVidia [43, 42] et OpenCL par Khronos groupe [82], facilitent leur programmation et donc contribuent à leur utilisation croissantes.

Karas *et al.* [104] proposent d'utiliser l'algorithme développé dans ce chapitre pour calculer des ouvertures et des granulométries sur une carte graphique avec des orientations quelconques. Chaque ligne 1-D est affectée à un thread de la carte graphique et plus il y a de lignes à calculer et plus les gains seront importants : de  $\times 5$  jusqu'à  $\times 20$  pour de grandes images avec une carte graphique de dernière génération.

## 2.8 Bilan et perspectives

Ce chapitre présente un nouvel algorithme 1-D flexible et rapide permettant de calculer des ouvertures, des ouvertures par reconstruction, des granulométries ou des arbres de composantes connexes dont voici les caractéristiques principales :

1. General-Purpose Processing on Graphics Processing Units

## 2. ÉLÉMENT STRUCTURANT INVARIANT PAR TRANSLATION

---

- la complexité théorique est constante par pixel,
- la gestion des bords peut être réalisée de deux façons différentes,
- n'importe quel type d'entrée peut être utilisé ; des entiers, des flottants de toutes précisions ou plus généralement, des valeurs qui forment un groupe ordonné,
- l'application à des images 2-D, ..., n-D est facile puisque il est toujours possible d'extraire des lignes 1-D de ce volume,
- cet algorithme est le plus rapide pour le calcul des ouvertures 1-D (à l'exception des données 8-bits où l'algorithme de Van Droogenbroeck reste devant),
- aucun algorithme de l'état de l'art ne rivalise avec celui proposé pour le calcul des granulométries,
- la parallélisation de l'algorithme sur le CPU est triviale et permet d'obtenir des accélérations de l'ordre de 433% pour un processeur i7 pour des ouvertures horizontales,
- la parallélisation sur la carte graphique permet d'atteindre des accélérations de l'ordre de 500% à 2000% pour des images de très grandes tailles.

Pour étendre l'utilisation de cet algorithme, on s'intéresse au calcul des granulométries locales ; en chaque élément du signal 1-D, on peut définir un histogramme qui stockera le volume de toutes les cordes auxquelles l'élément courant appartient. Cela nous permettra d'obtenir des descripteurs en vue de réaliser une analyse statistique et un classement point à point. Enfin, en introduisant un nouveau balayage de l'image, plus nécessairement linéaire, on peut définir un nouvel opérateur permettant de calculer efficacement des ouvertures par chemins. Cet opérateur est introduit dans le chapitre suivant (chapitre 3).

Pour les arbres de composantes connexes, une extension consisterait à construire un arbre pour chaque signal 1-D que l'on a extrait à partir de l'image initiale,  $\{f_{\alpha,k}\}_{k \in K}$  pour obtenir un ensemble d'arbres. Cette *forêt* permettrait de reconstituer l'arbre de composantes connexes 2-D associé à cette image, en utilisant l'algorithme de Wilkinson *et al.* [252] ou de Matas *et al.* [135, 134] pour fusionner plusieurs arbres ensemble. L'algorithme développé dans ce chapitre serait alors une étape d'un nouveau processus permettant de construire l'arbre de composantes connexes 2-D d'une image.

---

## Filtre par chemins : nouvel opérateur permettant de calculer des ouvertures parcimonieuses par chemins

---

*«Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.»*  
 Antoine de Saint-Exupéry, Terre des hommes ch.III, 1939

Les ouvertures par chemins maximaux (PO<sup>1</sup>) sont des opérateurs morphologiques permettant de préserver les structures longues et fines d'une image en niveaux de gris. Ces opérateurs parcourent l'ensemble des chemins d'une image et les filtrent avec un critère de taille. Le nombre de chemins à explorer et la grande redondance de l'information expliquent la relative lenteur de ces méthodes.

On introduit dans ce chapitre un nouvel opérateur morphologique, les *ouvertures parcimonieuses par chemins* (PPO<sup>2</sup>) qui permettent de résoudre le problème des temps de calcul. Cet opérateur n'explore qu'un nombre restreint de chemins parmi ceux considérés par les ouvertures par chemins classiques. En sélectionnant astucieusement ce sous ensemble, les résultats obtenus sont similaires à ceux des ouvertures par chemins cependant, les temps de calcul sont considérablement réduits. L'algorithme présenté dans le chapitre précédent (chapitre 2) constitue le socle de cet opérateur et est utilisé pour analyser tous les chemins 1-D. Il permet donc de construire des PPOs pour n'importe quel type de données d'entrée, indépendamment de la longueur de l'ouverture considérée.

La précision de cet opérateur est étudiée puis améliorée pour résoudre un problème classique des ouvertures par chemins, l'invariance par rotation et la surestimation des chemins diagonaux.

Les ouvertures parcimonieuses par chemins sont aussi étendues aux chemins incomplets ; chemin discontinu, corrompu par du bruit. Cet opérateur propose des temps de calcul qui restent insensibles à la taille de l'ouverture et à la tolérance considérée, tout en étant robuste aux bruits.

Ces opérateurs parcimonieux atteignent une accélération de plusieurs ordres de grandeur par rapport aux ouvertures par chemins classiques. On donne l'exemple des PPOs pour des chemins incomplets où les temps de calcul sont passés de plusieurs minutes à quelques millisecondes, tout en préservant la qualité du filtrage.

---

1. PO : Path Opening en anglais  
 2. PPO : Parsimonious Path Opening en anglais

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

---

Mots clés : Ouvertures parcimonieuses par chemins, morphologie mathématique, chemins complets, chemins incomplets, ouvertures par chemins, chemins localement maximaux, invariance par rotation.

#### 3.1 État de l’art

Les ouvertures par chemins sont des opérateurs récents en morphologie mathématique. Ils ont été introduits en 2004 par Heijmans *et al.* [89, 90], pour détecter les structures filiformes qui ne sont pas nécessairement rectilignes. Ces opérateurs utilisent un graphe sous-jacent qui est orienté et sans cycle afin de définir un ensemble de chemins qui seront analysées et comparées à un paramètre de longueur. Tous les chemins ayant une taille inférieure à ce seuil seront supprimés. Cela permet d’obtenir un filtrage efficace avec un critère de taille, afin de détecter les structures fines de l’image.

Cet opérateur est utilisé dans la détection des routes à partir d’images de satellites [238, 107], dans le filtrage des vaisseaux sanguins [226], mais aussi dans la caractérisation de milieux fibreux [38] ou dans la détection des défauts de surfaces de pièces métalliques, chapitre 7.

Les temps de calcul de l’implémentation de base des ouvertures par chemins sont importants. Aussi, Appleton et Talbot [5] ont défini un algorithme permettant de réduire considérablement les temps de calcul pour des images en niveaux de gris. Dans le même but, Luengo Hendriks [126] a défini en 2010, un algorithme rapide permettant d’étendre les ouvertures par chemins à des images à  $n$  dimensions. Néanmoins, même si les temps ont été considérablement améliorés, ces implémentations restent encore trop lentes pour un nombre important d’applications.

Outre les temps de calcul qui restent parfois prohibitifs pour certaines applications, le manque de robustesse par rapport au bruit est aussi un inconvénient majeur. Du fait de la nature filiforme des structures que l’on souhaite préserver, le bruit a un impact majeur et peut déconnecter certaines structures. Pour répondre à cette problématique, Talbot et Appleton [224] développèrent un algorithme pour les chemins incomplets. Une tolérance est insérée permettant de s’abstraire des pixels corrompus par le bruit. Cependant, les temps de calculs augmentent d’une manière considérable limitant ainsi l’utilisation de cet opérateur en pratique.

Enfin, le dernier inconvénient des ouvertures par chemins est le manque de précision. En effet, cet opérateur n’est pas invariant par rotation et suivant certaines orientations, on observe une grande variabilité sur la mesure de la taille des structures.

La motivation première de ce travail est de proposer un algorithme très rapide, robuste aux bruits et permettant d’extraire avec précision les structures fines de l’image. Les ouvertures parcimonieuses par chemins permettent de répondre en tous points à ces besoins. L’organisation de ce chapitre est la suivante : la section 3.2 résume le fonctionnement des ouvertures par chemins classiques. Les sections 3.3, puis 3.4, introduisent les ouvertures parcimonieuses par chemins, pour des chemins complets et incomplets. Enfin, nous verrons quelques considérations pratiques pour obtenir un algorithme rapide, (section 3.5), puis nous étudierons la précision, la complexité, les temps de calcul (sections 3.6 et 3.8). Ce chapitre est illustré par deux exemples d’applications : la détection des défauts de surface de pièces métalliques et la mesure des fibres d’élastine du derme de la peau (section 3.7).

#### 3.2 Notions de base sur les ouvertures par chemins

Les ouvertures par chemins ont été introduites pour offrir une plus grande flexibilité par rapport au supremum des ouvertures par des segments (chapitre 2), puisqu’elles peuvent analyser des structures

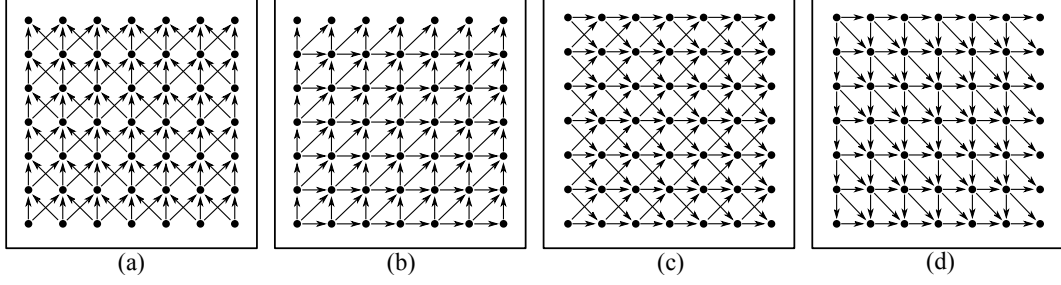


FIGURE 3.1 – Exemple de graphes communément utilisés pour les ouvertures par chemins. Ils définissent une orientation principale : (a) chemins sud nord (**S-N**), (b) **SO-NE**, (c) **O-E** et (d) **NO-SE**. Les graphes (b) (c) et (d) sont des rotations du graphe (a).

filiformes qui ne sont pas parfaitement linéaires. Cette section fournit quelques rappels sur les définitions et les caractéristiques de cet opérateur.

### 3.2.1 Graphe orienté acyclique

Soit  $I$  une image binaire, une application de  $D$  dans  $\{0,1\}$  avec  $D$  un sous domaine rectangulaire de  $\mathbb{Z}^2$ . On définit  $X$  comme le sous ensemble de  $D$  où l'image binaire vaut 1. On équipe  $D$  avec un graphe  $G$  permettant de définir un ensemble de chemins. Dans la suite, nous considérons des graphes acycliques orientés aussi appelés hiérarchies ou DAG (de l'Anglais directed acyclic graph). Un tel graphe peut être décrit comme une fonction d'un ensemble  $D$  (les nœuds) dans  $\mathcal{P}(D)$ . Pour tous les couples de points  $x$  et  $y$  appartenant à  $D$ , on dit que  $x$  est lié à  $y$  si et seulement si  $y \in G(x)$ . On définit le graphe inverse, noté  $G^- : D \rightarrow \mathcal{P}(D)$  et pour tout  $x$  appartenant à  $D$ ,  $y \in G^-(x)$  ( $x$  est lié à  $y$  par le  $G^-$ ) si et seulement si  $x \in G(y)$  ( $y$  est lié à  $x$  par  $G$ ). On note  $G_X$ , le graphe restreint à  $X$  et la figure 3.1 regroupe les graphes communément utilisés pour les ouvertures par chemins.

Un chemin est introduit de la manière suivante :

#### DÉFINITION 3.1

Une séquence  $\pi = (x_1, x_2, \dots, x_n), n > 0$ , de points dans  $X$  est un chemin de  $G_X$  si et seulement si  $\forall i, 1 \leq i \leq n-1, x_{i+1} \in G_X(x_i)$  ( $x_i$  est lié à  $x_{i+1}$  par  $G_X$ ). Sa longueur est définie par le nombre de points de cette séquence.  $x_1$  est appelé point de départ du chemin.

Soit  $\lambda : D \rightarrow \mathbb{N}$ , une carte de distance qui stocke la longueur maximale du plus grand chemin passant par chaque point. Pour un graphe acyclique, orienté et fini  $G_X$ , on notera  $\Pi(G_X)$  l'ensemble fini de tous les chemins de  $G_X$ .

### 3.2.2 Ouvertures par chemins binaires

Calculer la longueur maximale des chemins nécessite uniquement deux parcours avec les graphes  $G_X$  et  $G_X^-$ . Ces deux graphes sont représentés en noir sur les figures 3.2b et 3.2c, pour une image binaire donnée (figure 3.2a). On associe à chacun des deux graphes une carte de distance  $\lambda^+$  et  $\lambda^- : D \rightarrow \mathbb{N}$ , pour calculer la longueur des chemins à travers ces graphes  $\forall x \in X$  :

$$\lambda^+(x) = \max_{y \in G(x)} \lambda^+(y) + 1, \quad (3.1)$$

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

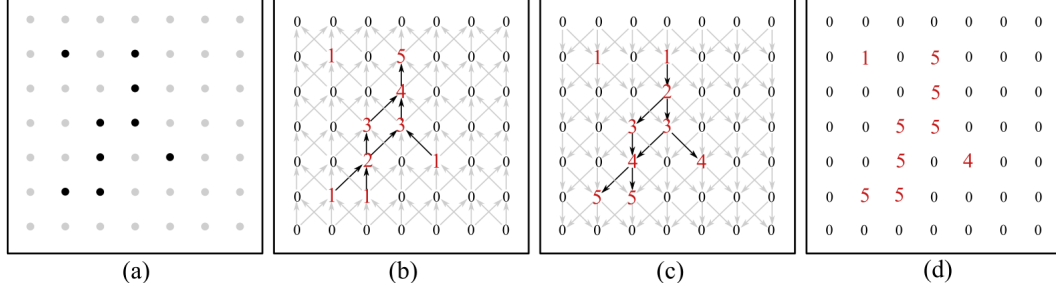


FIGURE 3.2 – Exemple de calcul des ouvertures par chemins : (a) image binaire initiale, (b)  $\lambda^+$  distances montantes grâce au graphe  $G_X$  (représenté en noir), (c)  $\lambda^-$  distances descendantes grâce au graphe  $G_X^-$  et (d) la carte des distances  $\lambda$  que l'on peut binariser pour obtenir une ouverture par chemins.

$$\lambda^-(x) = \max_{y \in G^-(x)} \lambda^-(y) + 1. \quad (3.2)$$

Les valeurs de  $\lambda^+$  et de  $\lambda^-$  sont représentées sur les figures 3.2b et 3.2c.

Enfin,  $\lambda$  est calculée simplement en sommant les deux cartes de distances et en soustrayant 1 :

$$\lambda(x) = \lambda^+(x) + \lambda^-(x) - 1. \quad (3.3)$$

Pour un pixel donné  $x$ , la valeur de  $\lambda(x)$  est égale à la longueur du plus grand chemin (vertical) passant par lui. Binariser  $\lambda$  avec un seuil égal à  $L$  permet de conserver uniquement les chemins plus grands que  $L$ . Cette opération correspond à une ouverture, notée  $\Gamma_L^{PO}(X)$  :

$$\Gamma_L^{PO}(X) = \{x \in D \mid \lambda(x) \geq L\} \quad (3.4)$$

Une ouverture par chemins, utilisant un graphe et son inverse, permet de filtrer les chemins dans une unique orientation. Il s'agit donc d'un filtre directionnel. On rappelle que le supremum des ouvertures est une ouverture. Ainsi, un traitement 2-D isotrope est obtenu en calculant le supremum des ouvertures par chemins en utilisant des rotations du graphe original (la figure 3.1 permet de visualiser l'ensemble des graphes pour caractériser les structures dans les 4 directions principales : verticale, horizontale et les deux diagonales). On appelle  $\Pi_f$ , l'ensemble de tous les chemins de l'image, associés aux différents graphes utilisés et restreint au domaine  $D$  de  $f$ .

#### 3.2.3 Ouvertures par chemins en niveaux de gris

Soit  $f : D \rightarrow V$  une image en niveaux de gris avec  $V = \mathbb{Z}$  ou  $\mathbb{R}$ , l'ensemble des valeurs et soit  $X^h = \{x \mid f(x) \geq h\}$  la binarisation de  $f$  au niveau  $h$ . L'ouverture par chemins  $\gamma_L^{PO}$  est idempotente, anti-extensive et croissante et, de la même manière que dans la section 2.2.1, l'ouverture commute avec l'opérateur de binarisation, permettant d'étendre cet algorithme à des images en niveaux de gris :

$$\gamma_L^{PO}(f) = \vee \{h \in V \mid x \in \Gamma_L^{PO}(X^h(f))\}. \quad (3.5)$$

L'utilisation de l'équation 3.5 pour étendre cet algorithme est facile mais n'est pas utilisée en pratique. En effet, Appleton et Talbot proposèrent une mise à jour rapide et efficace des cartes de distance  $\lambda^+$  et  $\lambda^-$  afin d'accélérer les temps de calcul de cet opérateur. Nous ne décrivons pas les optimisations utilisées et nous dirigeons le lecteur soit vers [5], soit vers [126].

Même si les temps de calcul ont été considérablement accélérés, les ouvertures par chemins restent des opérateurs relativement lents, et nous allons voir comment résoudre ce problème dans la section suivante.

Dans le reste de ce chapitre, nous présenterons et illustrerons ce travail uniquement avec la détection de structures claires dans l'image. En effet, les fermetures par chemins (les fermetures parcimonieuses par chemins) sont calculées exactement de la même manière, en prenant des POs (des PPOs) sur l'inverse de l'image initiale.

### 3.3 Ouvertures parcimonieuses par chemins

L'idée principale des ouvertures parcimonieuses par chemins est d'analyser un ensemble restreint de chemins au lieu de tous les explorer. En effet, le nombre de chemins augmente exponentiellement avec la taille de l'image et parmi eux, seulement quelques-uns apportent une information utile. L'objectif clé des ouvertures parcimonieuses par chemins est donc de déterminer ce sous ensemble en définissant un balayage adapté et rapide.

On donne tout d'abord quelques propriétés sur ces opérateurs puis on définit la stratégie d'extraction d'un bon sous-ensemble de chemins pour calculer des ouvertures parcimonieuses.

#### 3.3.1 Filtrage des chemins

On rappelle que  $\Pi_f$  est l'ensemble des chemins d'une image en niveaux de gris, associé à plusieurs graphes  $G$  orientés dans plusieurs directions de l'espace et restreint au support  $D$  de l'image.

##### DÉFINITION 3.2

Soit  $f : D \rightarrow V$  une image en niveaux de gris.  
 Soit  $G$  un graphe acyclique, orienté, sur  $D$ .  
 Soit  $\pi$  un chemin de  $G$  appartenant à l'ensemble  $\Pi_f$ .  
 $\gamma_L^\pi(f)$  correspond alors à une ouverture 1-D de  $f$ , de taille  $L$ , le long du chemin  $\pi$ .

Cet opérateur travaille sur un signal 1-D et possède les trois propriétés fondamentales d'une ouverture : il est anti-extensif, croissant et idempotent. Lorsque l'on considère tous les chemins possibles de l'image, le supremum des ouvertures reste une ouverture et l'équation 3.5 est équivalente à :

$$Eq.(3.5) \Leftrightarrow \gamma_L^{PO}(f) = \bigvee_{\pi \in \Pi_f} \gamma_L^\pi(f). \quad (3.6)$$

Pour des ouvertures parcimonieuses par chemins, on considère maintenant  $\Pi$ , le sous-ensemble de chemins, tel que  $\Pi \subset \Pi_f$  et  $Card(\Pi) \ll Card(\Pi_f)$ . La définition 3.2 propose de filtrer un chemin avec une ouverture mono-dimensionnelle. Néanmoins, lorsque l'on considère uniquement un sous ensemble de chemins de l'image, le supremum des ouvertures peut ne pas être une ouverture. En effet, la propriété d'idempotence n'est vérifiée que si l'ensemble des chemins est fixé ou indépendant de l'image  $f$ . Aussi, lorsque l'on combinera plusieurs filtres parcimonieux sur une même image, on utilisera toujours le même sous ensemble de chemins  $\Pi$ . Les ouvertures parcimonieuses par chemins sont donc définies de la manière suivante :

$$\gamma_L^{PPO}(f) = \begin{cases} \bigvee_{\pi \in \Pi} \gamma_L^\pi(f) & \text{si } x \in \pi, \\ 0 & \text{sinon.} \end{cases} \quad (3.7)$$



### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

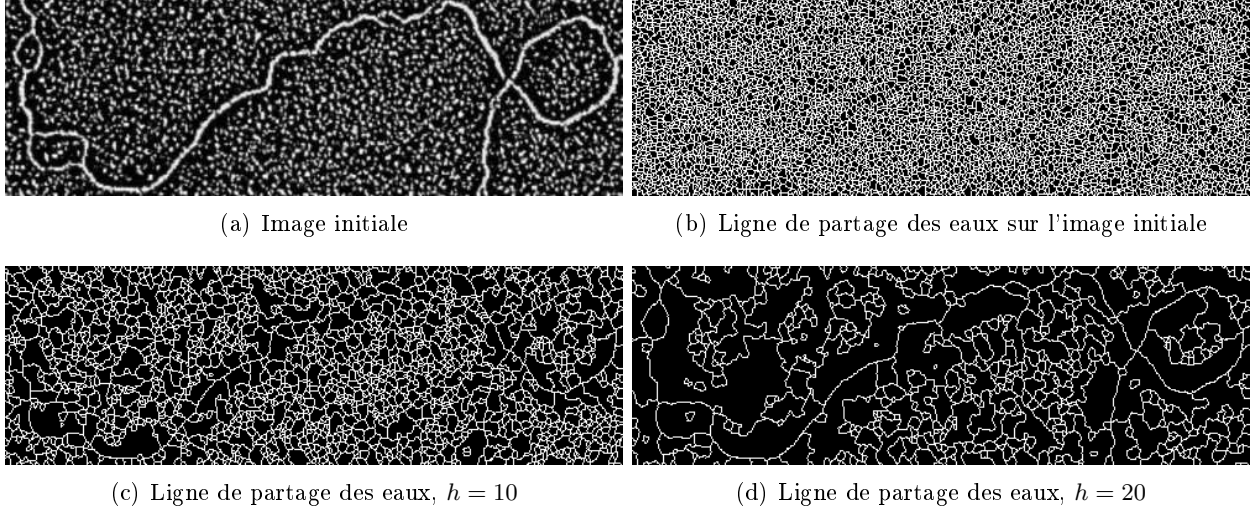


FIGURE 3.3 – Extraction d'un sous ensemble de chemins avec la méthode LPE : (a) image initiale, (b) LPE sur l'image initiale, (c) LPE sur une image simplifiée avec les  $h$ -minima ( $h = 10$ ) et (d)  $h = 20$ .

Les PPO sont donc bien des ouvertures, où tous les pixels de  $f$  qui ne font pas partie d'un chemin  $\pi \in \Pi$  sont initialisés à 0. La section suivante montre plusieurs méthodes pour extraire ce sous-ensemble de chemins.

#### 3.3.2 Extraction des chemins

Obtenir des informations sur une structure nécessite que cette dernière soit au moins parcourue par un chemin. Ainsi, différentes stratégies sont développées dans cette section pour extraire le sous ensemble de chemins le plus représentatif des structures de l'image. Une PPO sera calculée de deux manières différentes, soit en réduisant le support de l'image à un sous domaine judicieusement choisi sur lequel on calculera des ouvertures par chemins classiques, soit en analysant directement chaque chemin de ce sous ensemble avec l'équation 3.7.

##### Stratégie utilisant la ligne de partage des eaux (LPE)

Les structures claires qui doivent être détectées peuvent être vues comme des lignes de crêtes. Pour obtenir des chemins sur ces lignes de crêtes, on peut calculer la ligne de partage des eaux [18, 247] directement sur l'image initiale.

Soit  $w : D \rightarrow \{0,1\}$ , une image binaire correspondant à la ligne de partage des eaux relative à  $f$ . On définit  $f' : D' \rightarrow V$ ,  $D' \subset D$ , comme la restriction de  $f$  à  $D' = \{x | w(x) = 1, x \in D\}$  et  $f'(x) = f(x)$  pour tout  $x$  appartenant à  $D'$ . Une ouverture parcimonieuse par chemins de taille  $L$  utilisant la ligne de partage des eaux est alors définie par l'équation suivante :

$$\gamma_L^{PPO-LPE}(f) = \begin{cases} \gamma_L^{PO}(f') & \text{si } x \in D' \\ 0 & \text{sinon} \end{cases} \quad (3.8)$$

Les chemins inclus dans le support  $D'$ , sont obtenus en utilisant la définition 3.1 avec un graphe restreint à  $D'$ . La figure 3.3(b) présente la ligne de partage des eaux  $w$  associée à la figure 3.3(a), une

image d'ADN observée au microscope électronique à balayage. Le sous ensemble de chemins obtenu est parcimonieux et nous avons généralement  $\text{card}(D') \ll \text{card}(D)$ , 3.3(b). Ici, la ligne de partage des eaux est calculée directement sur  $f$ , à partir de tous ses minima. Néanmoins, la sur-segmentation peut être contrôlée avec des marqueurs, afin d'obtenir une ligne de partage des eaux moins dense. On montre les résultats de cette transformation sur la même image initiale, cette fois ci simplifiée par un filtre  $h$ -minima [244], figure 3.3(c) et 3.3(d).

Le nombre de chemins et par conséquent les temps de calcul, sont liés au contenu de l'image. Limiter la sur-segmentation accélère les PPOs puisque  $f'$  possèdera un domaine très restreint. Néanmoins, avant de calculer une ouverture par chemins sur  $f'$ , il est nécessaire d'utiliser au préalable un opérateur de ligne de partage des eaux. Bien que ce soit un opérateur relativement rapide, cette étape demande un temps de calcul important qui, au final, ne permet pas de réduire les temps de calcul des ouvertures parcimonieuses par chemins par rapport aux ouvertures par chemins classiques.

À cet inconvénient majeur vient s'ajouter un nouveau : il n'est pas possible d'individualiser chaque chemin. De plus, comme le résultat de la LPE n'est pas contraint par la connectivité des graphes de la figure 3.1, le filtre PO peut déconnecter certains chemins lorsque ceux-ci sont trop tortueux (la figure 3.6 présente le résultat des PPOs avec la ligne de partage des eaux).

On propose donc d'autres stratégies pour extraire le sous ensemble de chemins.

#### Stratégie utilisant les chemins globalement maximaux (Glo)

Dans cette section, chaque chemin sera extrait et filtré individuellement. On sélectionne donc un graphe sous-jacent orienté et sans cycle (le graphe de la figure 3.1). Puis, on recherche un chemin optimal entre deux points de l'image. Si on visualise une image comme un relief topographique, un chemin optimal revient à trouver un chemin dans le graphe entre ces deux points, qui soit le plus haut possible en moyenne. L'algorithme de Dijkstra [52] permet de faire de telles recherches. Néanmoins, le graphe est acyclique et orienté ce qui permet d'utiliser des algorithmes spécifiques plus rapides, basés sur des approches liées à la programmation dynamique [14]. Ces algorithmes sont connus sous le nom d'*algorithmes de plus long chemin* [71, 31]. Nous pouvons donc définir un chemin optimal à partir de n'importe quel point de départ et pour tous les graphes de la figure 3.1. Pour avoir un sous ensemble parcimonieux de chemins dans une image, on utilisera par la suite, tous les pixels appartenant aux bords de l'image comme point de départ de trois chemins optimaux avec trois graphes différents tel que :

#### DÉFINITION 3.3

Soit  $f : D \rightarrow V$  une image en niveaux de gris.  
 Soit  $\Pi_f$  un ensemble de chemins associés à tous les graphes  $G$  utilisés et restreint à  $D$ .  
 Un chemin  $\pi^{Glo}$  est un chemin globalement optimal si et seulement si son point de départ appartient à un des bords de  $D$  et si on a :

$$\pi^{Glo} \in \operatorname{argmax}_{\pi \in \Pi_f} \left( \sum_{x_j \in \pi} f(x_j) \right) \quad (3.9)$$

Pour un même point  $x$  du bord de  $D$ , on peut avoir plusieurs chemins globalement optimaux démarrant de ce point. En pratique, le premier chemin sera sélectionné mais la probabilité qu'un tel cas de figure se présente est faible. Cela nous permet d'introduire l'ensemble  $\Pi^{Glo}$  :

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

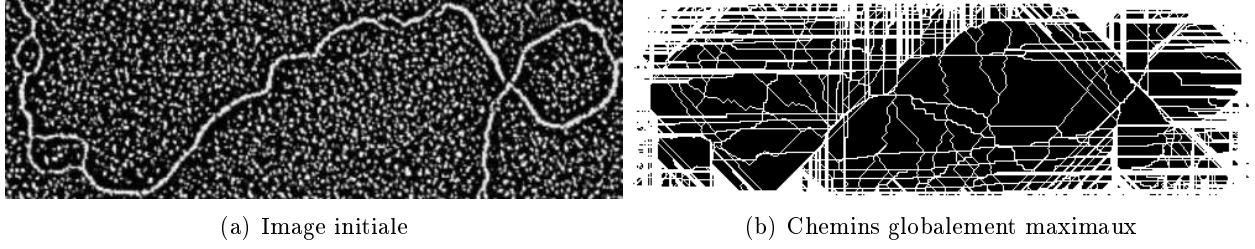


FIGURE 3.4 – Extraction du sous ensemble de chemins  $\Pi^{Glo}$  : (a) image initiale, (b) chemins globalement maximaux.

#### DÉFINITION 3.4

Un ensemble  $\Pi^{Glo} = (\pi_1^{Glo}, \pi_2^{Glo}, \dots, \pi_n^{Glo}) \subset \Pi_f$  est l'ensemble des chemins globalement optimaux de l'image  $f$ .

À partir du moment où chaque chemin est individualisé, les ouvertures parcimonieuses par chemins de taille  $L$ , utilisant des chemins globalement maximaux sont définis simplement par l'équation suivante :

$$\gamma_L^{PPO-Glo}(f) = \bigvee_{\pi \in \Pi^{Glo}} \gamma_L^\pi(f). \quad (3.10)$$

La figure 3.4(b) est une représentation  $\Pi_f^{Glo}$ . On remarque que le nombre de chemins de cet ensemble est faible, si on le compare à l'ensemble des chemins d'une image. Ils passent par les structures les plus brillantes et les plus longues de l'image et certaines zones n'ont aucun chemin les traversant. On qualifie ces régions d'*aveugles*, puisque qu'aucune information ne peut en être déduite. Avec cette stratégie d'extraction des chemins, la taille moyenne des régions aveugles est importante et potentiellement, des structures localisées dans ces zones et plus grandes que  $L$ , peuvent être supprimées à tort. Cependant, on remarque que les structures principales que l'on cherche à quantifier sont bien analysées.

Utiliser des chemins globalement optimaux pour extraire le meilleur sous ensemble permet d'individualiser les chemins et d'analyser des structures filiformes parmi les plus longues et les plus brillantes de l'image. Néanmoins, la recherche des chemins optimaux est très coûteuse en temps de calcul et la taille importante des régions aveugles engendre trop de faux négatifs. Pour corriger ces inconvénients, on développe une stratégie intermédiaire : *les chemins localement maximaux*.

#### Stratégie utilisant les chemins localement maximaux (Loc)

Cette stratégie combine les avantages des deux méthodes précédentes tout en limitant leurs inconvénients. On veut individualiser les chemins comme pour la méthode *Glo*, tout en limitant la taille des régions aveugles comme pour la méthode *LPE*.

Une fois de plus, on étend la définition 3.1 pour définir des chemins localement maximaux :

#### DÉFINITION 3.5

Soit  $D$  un domaine rectangulaire de  $\mathbb{Z}^2$  d'une image  $f$ .  
 Soit  $G$  un graphe acyclique et orienté sur  $D$ .  
 $\pi = (x_1, \dots, x_n)$  est un chemin localement maximal si et seulement si le point de départ appartient

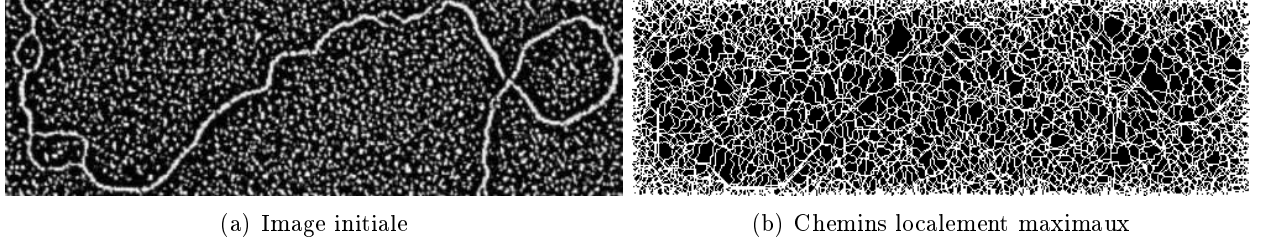


FIGURE 3.5 – Extraction d'un sous ensemble de chemins : (a) image initiale, (b) chemins localement maximaux.

aux bords de  $D$  et si  $\forall x_i, 0 \leq i < n$ , on a :

$$x_{i+1} \in \operatorname{argmax}_{x_j \in G(x_i)} \{f(x_j)\}. \quad (3.11)$$

L'équation 3.11 est utilisée pour construire un chemin itérativement à partir d'un point de départ  $x_1$ . Lorsqu'il n'y a plus de successeur à  $x_i$ , le chemin se termine. Enfin, si deux successeurs (ou plus) ont le même niveau de gris, alors l'orientation principale du chemin sera préservée en sélectionnant le pixel central.

De la même manière que pour la méthode *Glo*, chaque pixel touchant le bord de l'image est un point de départ de trois chemins. Cette stratégie permet de définir  $\Pi^{Loc}$ . Ce sous ensemble est représenté dans la figure 3.5(b) et on observe que la taille des régions aveugles est considérablement réduite par rapport à  $\Pi^{Glo}$ . Par conséquent, la probabilité d'analyser toutes les structures d'une taille supérieure à  $L$  augmente. De plus, la construction de ces chemins est très rapide puisque l'on ne fait que des recherches *locales*.

Les ouvertures parcimonieuses par chemins de taille  $L$ , utilisant des chemins localement maximaux, sont définies par l'équation 3.12 :

$$\gamma_L^{PPO-Loc}(f) = \bigvee_{\pi \in \Pi^{Loc}} \gamma_L^\pi(f). \quad (3.12)$$

Utiliser des chemins localement maximaux pour calculer des PPOs permet de réduire considérablement les temps de calcul tout en conservant une bonne précision. La figure 3.6 illustre l'extraction d'une molécule d'ADN à partir d'une image bruitée, en utilisant les ouvertures parcimonieuses par chemins pour les différentes méthodes d'extraction des chemins, *LPE*, *Glo* et *Loc*. On remarque tout d'abord que les détections sont minces et parcimonieuses par rapport aux ouvertures par chemins. De plus, on note que les résultats de ces filtres sont très similaires mais pas identiques : la détection avec  $\gamma_L^{PPO-LPE}$  est moins bonne que pour les autres méthodes. En effet, les chemins ne sont pas individualisés et peuvent être déconnectés en utilisant les graphes de la figure 3.1. On sous-estime donc la longueur réelle des structures.

La section suivante présente les ouvertures parcimonieuses sur des chemins incomplets.

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

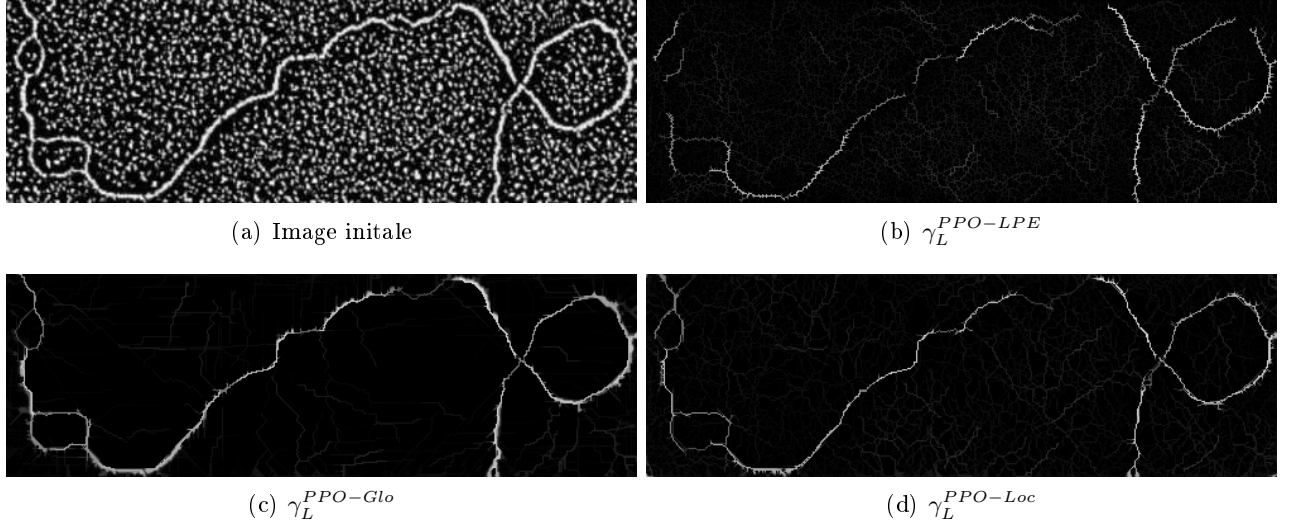


FIGURE 3.6 – Ouvertures parcimonieuses par chemins complets avec  $L = 40$ , pour les différentes stratégies d’extraction des chemins.

### 3.4 Ouvertures parcimonieuses pour des chemins incomplets

Talbot et Appleton, dans [224], firent une judicieuse remarque : “*Paths openings with increasing  $L$  are also increasingly sensitive to noise, as long paths are more likely to contain noisy pixels.*”. Le bruit peut déconnecter des longues structures filiformes avec comme conséquence, une sous-estimation de leurs longueurs réelles. Si on considère un signal 1-D d’une structure claire corrompue par du bruit (comme pour le signal de la figure 3.7(a)), alors les PPOs complets sous-estiment sa longueur en extrayant uniquement une partie du signal. À partir de cette observation, travailler avec des chemins incomplets semble indispensable lorsque les images sont bruitées. On introduit tout d’abord la notion de filtres alternés parcimonieux, puis les filtres alternés séquentiels parcimonieux, enfin les ouvertures parcimonieuses pour des chemins incomplets.

#### 3.4.1 Filtres parcimonieux alternés par chemins

Soit une image  $f$  avec une famille de chemins  $\Pi$ . D’après la définition 3.2,  $\forall \pi \in \Pi$ ,  $\gamma_L^\pi(f)$  est une ouverture de  $f$  de taille  $L$  le long du chemin  $\pi$ . En inversant ce signal, on définit l’opérateur complémentaire, une fermeture :

$$\varphi_L^\pi(f) = -\gamma_L^\pi(-f). \quad (3.13)$$

Dans la définition originale des ouvertures par chemins de Heijman *et al.* [90], inverser l’image et calculer des ouvertures revient à calculer des fermetures par chemins :  $\varphi_L^{PO}(f) = -\gamma_L^{PO}(-f)$ . Ce filtre modifie donc l’ensemble des chemins analysés et permet de filtrer des structures foncées sur fond clair. L’équation 3.13 agit différemment puisque le chemin  $\pi$  est positionné sur un chemin clair et calculer une fermeture le long de ce chemin permet de boucher les trous générés par le bruit. Les fermetures sur les chemins (équation 3.13) possèdent toutes les propriétés d’une fermeture (croissance, extensivité et idempotence) et du moment où ces opérateurs sont appliqués sur les mêmes chemins  $\Pi$ , on peut

composer les ouvertures et fermetures afin d'obtenir des filtres alternés par chemins :

$$\gamma\varphi_L^{PPO}(f) = \bigvee_{\pi \in \Pi} \gamma_L^\pi \left( \bigwedge_{\pi \in \Pi} \varphi_L^\pi(f) \right), \quad (3.14)$$

$$\varphi\gamma_L^{PPO}(f) = \bigwedge_{\pi \in \Pi} \varphi_L^\pi \left( \bigvee_{\pi \in \Pi} \gamma_L^\pi(f) \right). \quad (3.15)$$

#### 3.4.2 Filtres parcimonieux alternés séquentiels par chemins

Un filtre alterné séquentiel est un opérateur qui filtre l'image successivement avec des ouvertures et des fermetures de tailles croissantes. Aussi, on peut utiliser les filtres alternés définis dans la section précédente pour construire deux compositions de filtres alternés séquentiels (FAS) :

$$\gamma\varphi_L^{PPO-FAS} = \gamma\varphi_L^{PPO} \dots \gamma\varphi_2^{PPO} \gamma\varphi_1^{PPO}, \quad (3.16)$$

$$\varphi\gamma_L^{PPO-FAS} = \varphi\gamma_L^{PPO} \dots \varphi\gamma_2^{PPO} \varphi\gamma_1^{PPO}. \quad (3.17)$$

La figures 3.8(c) et 3.8(d) présentent les résultats des filtres parcimonieux alternés et alternés séquentiels par chemins sur l'image de la figure 3.5(a). On remarque que les filtres FAS proposent un filtrage moins brutal que les filtres alternés. Ainsi, la détection de la structure filiforme est considérablement améliorée avec les FAS par chemins parcimonieux. On note, néanmoins, que le bruit est aussi plus rehaussé avec des FAS. La figure 3.20 présente un deuxième exemple de filtrage pour l'extraction de structures filiformes. Enfin, ces opérateurs sont illustrés sur des défauts de surface de type fissure (voir l'annexe III).

#### 3.4.3 Ouvertures parcimonieuses pour des chemins incomplets

Outre les filtres alternés et les filtres alternés séquentiels par chemins parcimonieux, on peut définir un nouvel opérateur permettant de réaliser des ouvertures par chemins incomplets. Il faut donc reconnecter les chemins avec une fermeture de petite taille, avant de les filtrer avec une ouverture de taille  $L$ . On définit alors une PPOs pour des chemins incomplets (notée  $\gamma_{L,tol}^{PPO}(f)$ ) qui est donnée par l'équation suivante :

$$\gamma_{L,tol}^{PPO}(f) = \bigwedge \left( f, \bigvee_{\pi \in \Pi} \gamma_L^\pi (\varphi_{tol+1}^\pi(f)) \right), \quad (3.18)$$

avec  $tol$  une tolérance représentant la taille maximale des trous à combler et  $\varphi_{tol+1}^\pi$  est défini suivant l'équation 3.13. On remarque que l'infimum entre l'image originale et le résultat du filtrage est nécessaire pour préserver la propriété d'anti-extensivité de ce filtre.

La figure 3.7 illustre une ouverture, une fermeture et un PPO incomplet sur un signal 1D. La détection de la structure claire corrompue par du bruit a été rendue possible par l'utilisation de l'opérateur développé dans cette section. Un exemple 2-D est donné dans la figure 3.8, où l'on calcule des ouvertures parcimonieuses par chemins complets et incomplets ( $tol = 2$ ) sur l'image d'ADN avec des chemins localement maximaux. On constate ainsi que la détection de cette molécule est bien meilleure puisqu'elle est mieux préservée par ce filtre. Néanmoins, le bruit est aussi reconnecté et des structures longues sont trouvées dans le fond de l'image.

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

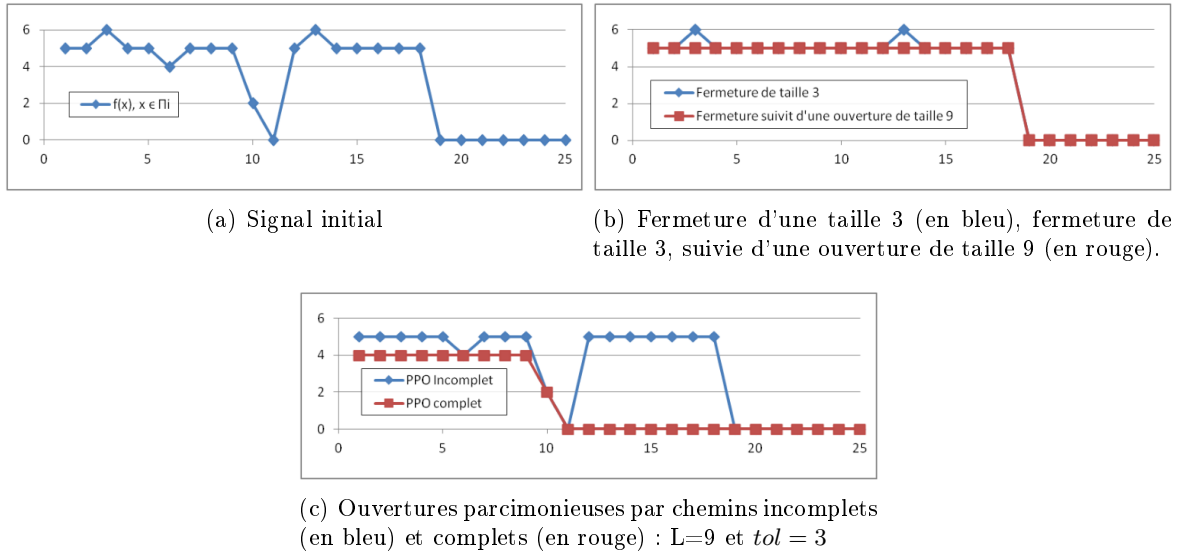


FIGURE 3.7 – Exemple du fonctionnement des ouvertures parcimonieuses par chemins complets et incomplets.

**REMARQUE :** la stratégie d'extraction des chemins *Glo* est plus robuste au bruit que la méthode *Loc*. En effet, la poursuite d'une structure claire, discontinue dans un environnement bruité peut échouer avec des chemins qui sont *localement* maximaux. Le choix du paramètre  $tol$  doit prendre en compte la quantité de bruit ainsi que le nombre de pixels à reconnecter. Choisir une grande valeur pour ce paramètre augmente le risque de reconnecter le bruit et ainsi de préserver des structures non désirées dans l'image. Il faut trouver le bon compromis entre la préservation des structures et la reconnexion du bruit. En pratique,  $tol$  dépasse rarement 10 pixels mais cette valeur dépend de l'image ainsi que de l'application.

En conclusion, les ouvertures parcimonieuses par chemins complets ou incomplets sont des nouveaux opérateurs permettant d'extraire un sous ensemble de chemins représentatifs des structures de l'images en utilisant trois stratégies différentes, section 3.3. La méthode basée sur la ligne de partage des eaux ne permet pas d'individualiser les chemins ni de réduire les temps de calcul, notre objectif premier. Les deux autres méthodes individualisent les chemins et simplifient le problème de filtrage en calculant des ouvertures et des fermetures sur des signaux 1-D. Cependant, seule la méthode sur les chemins localement maximaux permet d'extraire les chemins d'une manière rapide et efficace tout en limitant la taille des régions aveugles. On focalise donc notre attention sur les PPOs utilisant les chemins *Loc*, et l'on note alors  $\gamma_{L,tol}^{PPO} = \gamma_{L,tol}^{PPO-loc}$ . L'individualisation des chemins permet aussi d'améliorer la précision de la mesure de longueur de ces opérateurs. En effet, à cause de la discrétisation de la grille, la longueur de chaque liaison diagonale entre deux pixels voisins du chemin sera égale à  $\sqrt{2}$  ou plus généralement à  $\sqrt{dx^2 + dy^2}$  avec  $dx$  et  $dy$  le pas d'échantillonnage sur l'axe  $x$  et  $y$  respectivement. La section 3.6 étudie la précision de ces opérateurs et la section suivante donne des informations complémentaires.



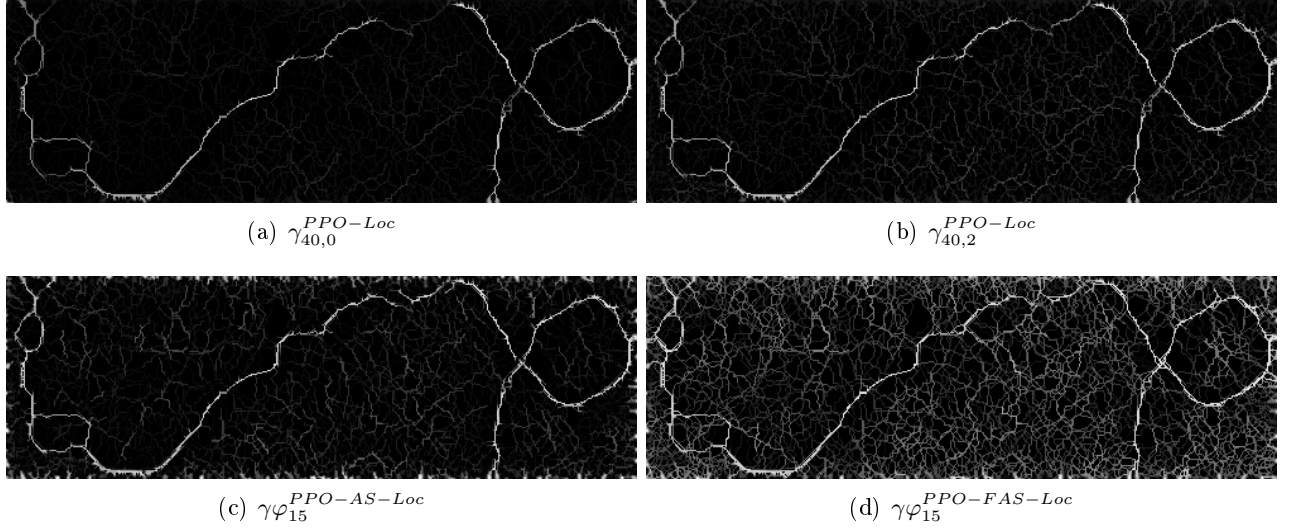


FIGURE 3.8 – Illustration des ouvertures parcimonieuses par chemins complets (a) et incomplets (b) avec des chemins localement maximaux ainsi que des filtres alternés (c) et alternés séquentiels (d).

### 3.5 Considérations pratiques

La vitesse de calcul est un élément important pour ces opérateurs. Aussi, cette section fournit des informations sur leur implémentation ainsi que sur une méthode pour réduire considérablement leur temps de calcul.

#### 3.5.1 Ouverture 1D en temps constant par pixel

Un chemin est une séquence de pixels qui est assimilé à un signal 1-D. Nous utilisons donc l'algorithme développé dans le chapitre 2 sur les ouvertures 1-D en temps constant par pixel, pour calculer  $\gamma_L^\pi$ . Le paramètre de taille des PPOs n'a donc plus d'influence dans la complexité algorithmique de cet opérateur et ceci constitue un avantage certain par rapport à la meilleure implémentation des ouvertures par chemins classiques.

La complexité est étudiée dans la section 3.8.1 et on introduit dans la section suivante, un paramètre  $k$ , contrôlant la parcimonie de cet opérateur afin de réduire plus encore les temps de calcul.

#### 3.5.2 Paramètre de parcimonie $k$

En observant la figure 3.5(b), on remarque que beaucoup de chemins fusionnent très rapidement pour suivre les mêmes structures et ce, même s'ils n'ont pas le même point de départ. Ces chemins redondants n'apportent pas d'informations complémentaires. On peut donc aller plus loin dans la réduction du nombre de chemins en introduisant le paramètre  $k$ , contrôlant la parcimonie de cet opérateur ( $\gamma_{L,tol,k}$ ). Comme les chemins ayant des points de départ proches, ont une forte probabilité d'être attirés par la même structure,  $k$  est la distance en pixels entre deux points de départ. En utilisant ce paramètre, on divise par  $k$  le nombre total de chemins et on divise par la même occasion la complexité par  $k$ . Néanmoins, supprimer trop de chemins peut détériorer la qualité de la détection de cet opérateur. On étudiera dans la section 3.6, quelle est l'influence de ce paramètre sur les résultats.



### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

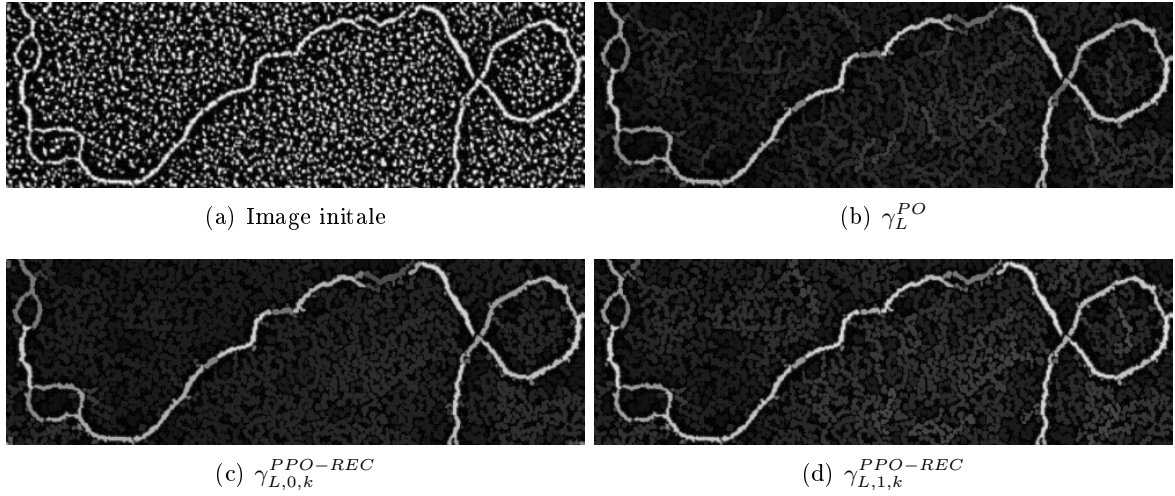


FIGURE 3.9 – Comparaison des ouvertures par chemins et des ouvertures parcimonieuses par chemins complets et incomplets avec  $L = 40$  et  $k = 1$ , avec une étape de reconstruction.

#### 3.5.3 Reconstruction morphologique

Les opérateurs définis précédemment sont des opérateurs parcimonieux. Les objets sont donc représentés par des structures filiformes. Néanmoins, si on souhaite avoir un résultat non parcimonieux, on peut reconstruire le résultat des PPOs sous l'image originale afin d'avoir une réponse en chaque pixel. Des implémentations rapides existent pour l'opérateur de reconstruction morphologique. Leur complexité est linéaire [244, 188] et par conséquent, même si l'application nécessite une étape de reconstruction, la complexité globale de cet algorithme ne change pas.

La figure 3.9 propose une comparaison entre les ouvertures par chemins, les PPOs complets et les PPOs incomplets. On constate que la détection est meilleure en utilisant des chemins incomplets, puisque l'on est capable de reconnecter les structures bruitées. Par rapport aux ouvertures par chemins classiques, la précision est très bonne et on propose de l'étudier dans la section suivante.

**REMARQUE :** Le choix des paramètres à utiliser pour les ouvertures parcimonieuses par chemins dépend des applications. Le paramètre de longueur  $L$  doit être choisit en fonction de la longueur des structures à préserver et le paramètre de tolérance  $tol$  par la taille des brèches le long des chemins. Enfin, pour choisir la valeur du paramètre de parcimonie  $k$ , il est nécessaire d'étudier la précision de cet opérateur.

### 3.6 Précision des ouvertures parcimonieuses par chemins

Utiliser uniquement un sous ensemble restreint de chemins peut avoir un impact sur la précision de ces opérateurs. Des régions aveugles apparaissent et des structures plus grandes que  $L$  peuvent être supprimées à tort. On propose dans cette section, de quantifier ces imprécisions et de comparer cette méthode avec des ouvertures par chemins classiques.

On verra que les ouvertures par chemins classiques sont elles aussi soumises à des imprécisions importantes en fonction de l'orientation des structures à analyser. L'invariance par rotation est une propriété essentielle pour un filtre, spécialement lorsque l'on calcule la distribution de taille des struc-

tures d'une image. On propose tout d'abord, d'analyser cette invariance par rotation pour les POs et les PPOs puis, on propose une méthode simple permettant d'améliorer la précision des filtres parcimonieux.

On étudiera par la suite, la précision des PPOs en fonction de la densité des structures dans l'image et du paramètre de parcimonie  $k$ . Enfin, nous comparerons les courbes granulométriques obtenues avec différents opérateurs. En effet, on peut définir des courbes granulométriques avec des ouvertures parcimonieuses par chemins afin d'obtenir une estimation de la longueur des structures de l'image. La définition de la distribution de taille est rappelée dans la section 2.2.2 et est résumée par l'équation suivante :

$$(PS(f))(l) = Meas(\gamma_l^{PPO-REC}(f) - \gamma_{l-1}^{PPO-REC}(f)), \quad l > 0, \quad (3.19)$$

avec  $Meas$  une mesure qui sera le volume. L'intégrale de la fonction  $PS$  donne la distribution de taille cumulée ou *courbe granulométrique* que l'on normalisera entre 0 et 1. Un exemple de courbe granulométrique est donné dans la figure 3.12.

Dans un souci de clarté et de simplicité, on prendra un pas d'échantillonnage identique sur l'axe  $x$  et sur l'axe  $y$  égale à 1. Cependant, il est facile d'étendre les éléments présentés ci-dessous à des images anisotropes.

#### 3.6.1 Invariance par rotation

L'invariance par rotation est une propriété importante pour les filtres et on montre dans cette section comment s'en approcher le plus possible avec des ouvertures parcimonieuses par chemins.

##### Description du problème

Une ouverture par chemins classique avec le graphe présenté dans la figure 3.1 surestime la longueur des chemins diagonaux. En effet, on cherche le chemin de longueur maximale et, avec un graphe diagonal, celui-ci peut zigzaguer dans la structure analysée. Une illustration de ce phénomène est proposée dans la figure 3.10a. Ce phénomène de zigzag est un problème bien connu des ouvertures par chemins maximaux que Luengo Hendriks [126] corrigea en partie. Il utilisa une idée initialement proposée par Buckley et Yang [31], qui consiste à ajouter une contrainte aux ouvertures par chemins. Deux graphes sont donc utilisés alternativement afin de définir la connectivité : un graphe de la figure 3.1 est associé à un graphe uniquement orienté dans sa direction principale. Dans l'exemple de la figure 3.10b, la longueur de la structure est mieux estimée pour un graphe diagonal. Néanmoins, il a été montré dans [126] que cet opérateur sous-estime quelque peu la longueur des chemins et que cette amélioration de la précision se fait au détriment des temps de calcul qui passent du simple au double. On appelle cette méthode *les ouvertures par chemins contraints* dans la suite de ce chapitre. Enfin, les PPOs pénalisent les liaisons diagonales par  $\sqrt{2}$ , ce qui permet d'avoir une bonne estimation de la longueur du chemin (figure 3.10c).

##### Expérience

Pour tester rigoureusement l'invariance par rotation de ces opérateurs, on génère un segment de droite horizontal de longueur et de largeur fixées à 40 et à 3 pixels respectivement. Avant de réaliser une rotation d'angle  $\alpha$ , on applique un filtre gaussien d'écart-type 2 à ce segment. Ce filtre gaussien est important puisqu'il limite les erreurs dues à la discrétisation de l'espace lors de la rotation [127]. On obtient donc une collection de fibres de même longueur mais avec des orientations différentes (figure

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

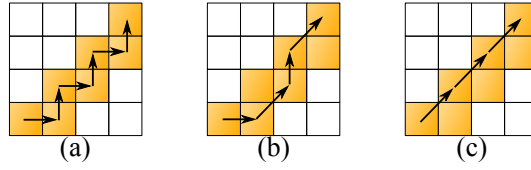


FIGURE 3.10 – Longueur du chemin avec un graphe diagonal orienté **SO-NE**. (a) PO classique (longueur du chemin : 7), (b) PO contraint (longueur du chemin : 5), (c) PPO (longueur du chemin :  $1 + 3\sqrt{2} = 5,24$ ).

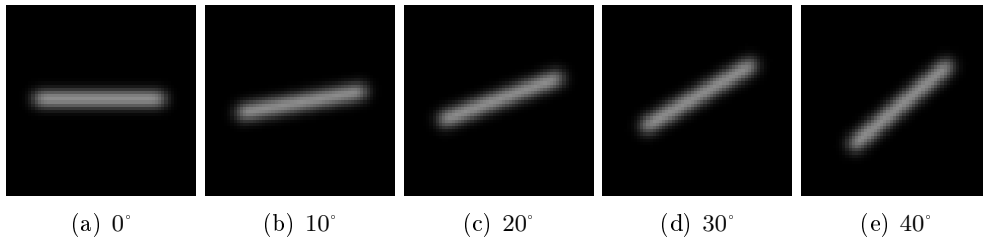


FIGURE 3.11 – Génération de fibres de longueur 40 pixels et de profil gaussien pour différentes orientations.

3.11). Idéalement, la courbe granulométrique cumulée est égale à 0 pour des longueurs inférieures à 40 pixels et à 1 pour des longueurs supérieures. Néanmoins, à cause du profil gaussien, de la largeur non unitaire des fibres et des imprécisions dans les mesures, des fluctuations vont apparaître. On calcule alors les courbes granulométriques pour les trois méthodes décrites (les POs, les POs contraints et les PPOs) ainsi que pour une quatrième méthode : le supremum des ouvertures par des segments (chapitre 2).

On constate que pour des orientations horizontales, les quatre méthodes fournissent la même estimation de la longueur de la fibre. Cependant, lorsque les fibres sont orientées suivant d'autres directions, des différences apparaissent (voir la figure 3.12). On constate, comme attendu, que les ouvertures par chemins classiques surestiment fortement la longueur des chemins diagonaux, et que les ouvertures par chemins contraints les sous-estiment. Ces deux opérateurs ne sont donc pas invariants par rotation comme peut l'être le supremum des ouvertures 1-D. La même conclusion peut être tirée des PPOs, puisqu'elles surestiment légèrement la taille des chemins pour certaines orientations. Ainsi, on propose dans la section suivante, d'améliorer la précision de cet opérateur.

#### Amélioration de la précision des PPO

Individualiser les chemins permet de mieux prendre en compte et de corriger l'effet introduit par la discrétisation du domaine. Pondérer les liaisons diagonales par  $\sqrt{2}$ , est une première étape dans l'amélioration de la précision des ouvertures par chemins parcimonieux. Cependant, l'expérience de la section 3.6.1 a montré que ce n'est pas suffisant. En effet, les PPOs surestiment la longueur des chemins diagonaux.

Pour améliorer la précision des PPOs et corriger l'effet de la discrétisation, on élargit le voisinage et on pondère les différentes liaisons en se servant des trois derniers pixels du chemin en cours de construction, au lieu des deux derniers. Pour rappel, le tableau 3.1 résume les deux configurations

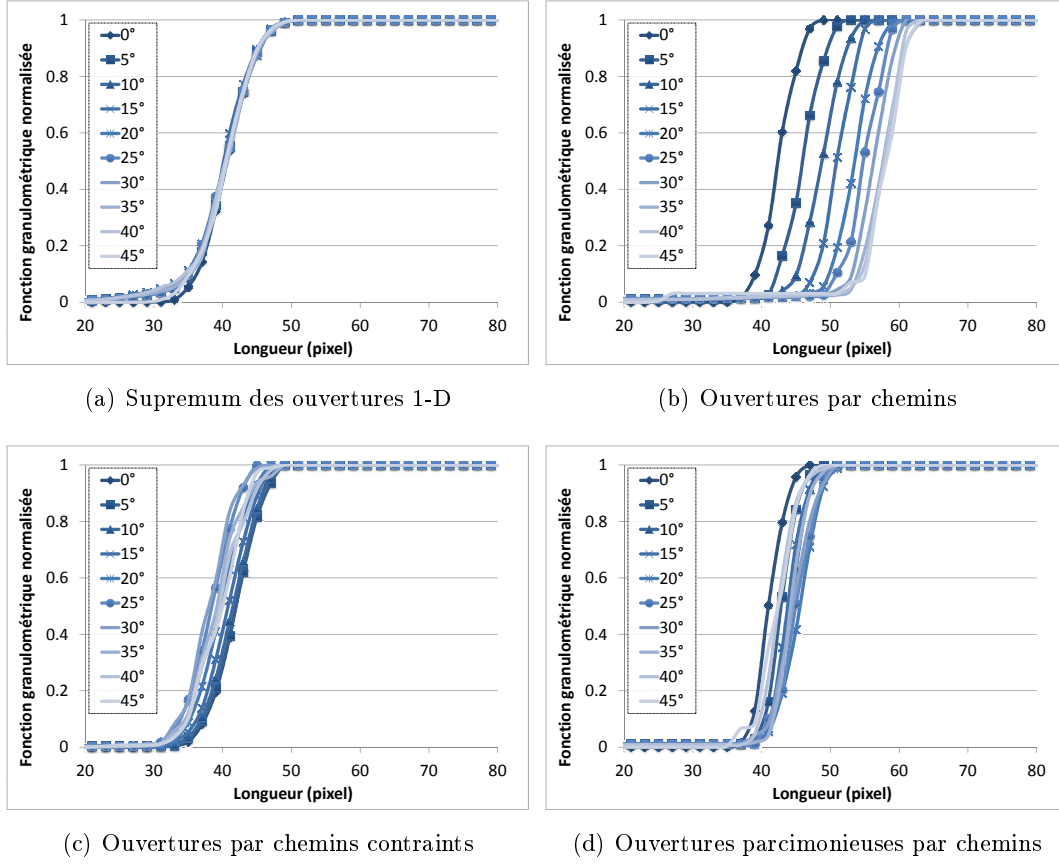


FIGURE 3.12 – Test de l'invariance par rotation. Les granulométries sont calculées avec différents types d'ouvertures.

possibles à deux points ainsi que leur pondération associée. Avec trois pixels, on dénombre au total 5 configurations différentes. Elles sont regroupées avec leur pondération, dans le même tableau 3.1 et on note que toutes les autres configurations peuvent être déduites des 5 premières avec de simples rotations.

En utilisant trois pixels au lieu de deux, l'estimation de la longueur des chemins diagonaux est meilleure. Cependant, d'un point de vue algorithmique, il est nécessaire de dissocier certaines configurations puisque l'on souhaite toujours parcourir et traiter le chemin **séquentiellement**. Ainsi, on ne connaît pas l'orientation future (et par conséquent, la configuration à utiliser) du chemin avant de traiter le pixel suivant. Ainsi, un paramètre est introduit permettant de stocker la dernière configuration utilisée (Pour l'initialisation du chemin ou sa réinitialisation, la dernière configuration est égale à 0 pour signifier qu'il faut utiliser l'une des deux configurations à deux points pour le prochain pixel). En fonction de la dernière configuration utilisée et de l'orientation du pixel suivant, on peut déduire la longueur de la liaison en pointillés dans les configurations du tableau 3.2.

- Pour les configurations 1 et 4, les longueurs des liaisons en pointillés doivent être respectivement égale à 1 et  $\sqrt{2}$  pour obtenir les distances totales de 2 et de  $2\sqrt{2}$ .

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

TABLEAU 3.1 – Présentation des configurations possibles et de leur longueur associée en utilisant deux pixels ou trois pixels.














	2 points		3 points				
Configurations							
Pondérations	1	$\sqrt{2}$	2	$\sqrt{5}$	$2\sqrt{2}$	$\sqrt{2}$	2

TABLEAU 3.2 – Présentation des différentes configurations ainsi que les longueurs associées. Les liaisons en pointillés (resp. en traits plein) correspondent à la configuration courante (resp. passée).

Numéro	Configuration	Longueur du segment en pointillé	Changement de configuration du passé
1		1	-
2		$\sqrt{5}-1$	0
3		$\sqrt{5}-\sqrt{2}$	0
4		$\sqrt{2}$	-
5		$\sqrt{2}-1$	3 ou 4
6		$2-\sqrt{2}$	1 ou 2

- Pour les configurations 2 et 3, les distances totales doivent être égales à  $\sqrt{5}$ . Pour atteindre cette valeur, les longueurs des chemins en pointillés doivent respectivement être égales à  $\sqrt{5}-1$  et à  $\sqrt{5}-\sqrt{2}$ . Une fois que l'on a obtenu cette configuration, on réinitialise le processus en changeant pour la configuration 0. Ainsi, le prochain pixel utilisera à nouveau l'une des deux configurations à 2 pixels.
- De la même manière, les longueurs des liaisons en pointillés doivent être égale à  $\sqrt{2}-1$  et à  $2-\sqrt{2}$  pour les configurations 5 et 6. Une fois cette combinaison terminée, on change la valeur de la dernière combinaison à 2 ou 3 pour qu'il constitue le début d'une autre configuration.

En utilisant une table de correspondance des longueurs en fonction des configurations, on peut facilement et sans surcoût, améliorer l'évaluation de la longueur d'un élément. Pour vérifier le résultat, on répète l'expérience de la section 3.6.1 et on constate que l'invariance par rotation est bien meilleur (figure 3.13).

#### 3.6.2 Précision des PPOs en fonction de la densité et du paramètre de parcimonie

Le contenu de l'image a une influence sur les résultats des PPOs puisque l'on n'utilise qu'un sous ensemble de chemins. Des régions aveugles apparaissent et engendrent des erreurs de mesure. On propose dans cette section, de mesurer l'impact de la densité et du paramètre de parcimonie  $k$  sur les résultats.

#### Génération d'un modèle aléatoire

Pour tester l'influence de la densité des images, il faut connaître exactement le contenu d'une image. Or nous ne pouvons pas utiliser les POs classiques ni les POs contraints comme références,

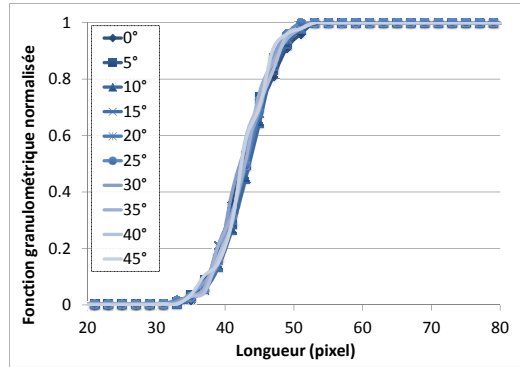


FIGURE 3.13 – L’invariance par rotation est testée en calculant des granulométries sur des fibres de même longueur mais avec des orientations différentes. Les ouvertures parcimonieuses par chemins sont des opérateurs invariants par rotation.

puisqu’ils sont eux aussi biaisés. Pour connaître exactement le contenu d’une image, pour faire plusieurs réalisations du même processus et pour modifier à volonté la densité des données, on utilisera une stratégie utilisant des modèles aléatoires [98, 99]. On prendra tout d’abord une fibre rectiligne, de taille comprise entre 3 et 80 pixels avec une orientation aléatoire, comme grain primaire. Le support de l’image aura une taille de 500 par 500 pixels et, pour connaître exactement le résultat de la courbe granulométrique, on ne tolère aucune intersection entre fibres ainsi qu’entre les fibres et le bord de l’image. Voici la méthodologie utilisée pour générer une réalisation de ce processus :

1. Pour une *fraction volumique* donnée, on estime le nombre de fibres à placer dans le modèle en utilisant une distribution de Poisson. La fraction volumique ( $V_V \in [0,1]$ ) est définie comme le rapport entre la surface occupée par les fibres et la surface de l’image.
2. On génère une fibre de longueur et d’orientation choisies aléatoirement (l’écart type du filtre gaussien est fixé à 2).
3. On insère cette fibre aléatoirement dans le modèle. Néanmoins, si aucune place respectant les contraintes de non intersection n’est trouvée, on choisira une autre fibre (retour à l’étape 2). Par conséquent, la densité demandée est sous-estimée puisque les petites fibres ont une probabilité plus grande de trouver une place dans le modèle. Cependant, ce phénomène n’a aucune importance puisque l’on calculera la densité réelle des fibres lorsqu’elles seront toutes insérées dans le modèle.
4. Une fois la fibre placée, on ajoute sa contribution à la courbe granulométrique théorique et on répète les étapes de 2 à 4 jusqu’à ce que toutes les fibres soient placées.

La figure 3.14 présente 4 réalisations différentes de ce processus pour des densités allant de 10% à 40%.

#### Influence de la densité des structures

Pour mesurer l’influence de la fraction volumique sur le calcul de la courbe granulométrique, on génère 100 modèles aléatoires (100 réalisations du même processus). Ainsi, on moyenne les courbes granulométriques expérimentales obtenues que l’on compare avec la courbe granulométrique moyenne théorique. En mesurant l’erreur relative entre ces deux courbes et en itérant ce calcul pour chaque densité retenue, on définit la courbe présentée dans la figure 3.15. Comme attendu, la corrélation

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

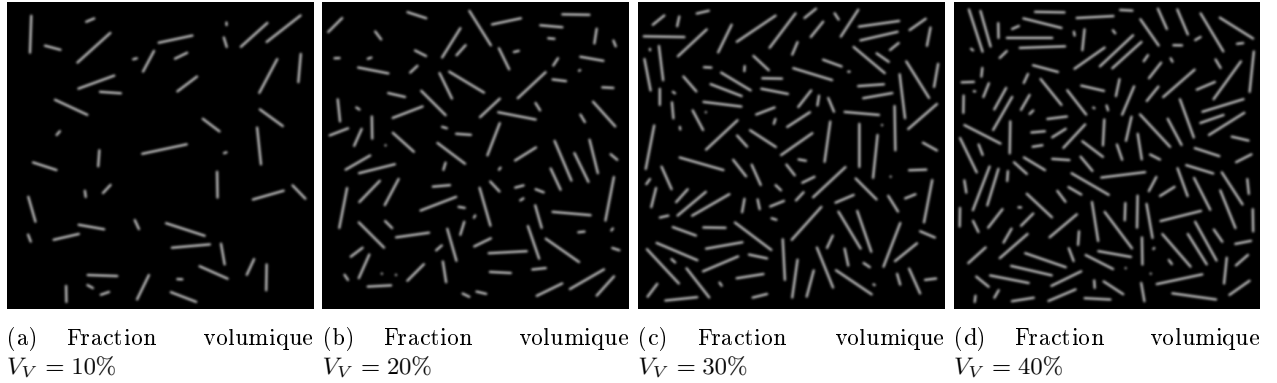


FIGURE 3.14 – Quatre réalisations du processus de génération de modèles aléatoires avec des fractions volumiques différentes.

entre l'erreur relative commise et la densité de l'image est positive. Plus la densité est forte et plus la probabilité de sous estimer la longueur d'une structure est importante. Pour une densité de 50%, l'erreur reste néanmoins faible et se situe au alentour des 11%. On observe principalement deux sources d'erreurs : les structures les plus longues n'ont pas toutes des chemins les traversant de part en part (leur longueur est sous-estimée) et, les petites structures localisées dans des régions aveugles ne sont pas prises en compte.

#### Influence du paramètre de parcimonie $k$

$k$  contrôle le nombre de chemins dans l'image. Plus  $k$  est élevé, plus le nombre de chemins est faible et plus la probabilité de commettre des erreurs augmente. Les temps de calcul sont donc fortement réduits mais en contrepartie, la qualité de la détection diminue. Pour tester l'influence de ce paramètre sur la précision des ouvertures parcimonieuses par chemins, on utilise le même procédé que pour la section précédente et on ajoute le calcul des courbes granulométriques moyennes avec différents  $k$ . La figure 3.15 présente les résultats et on observe que l'erreur relative commise augmente avec  $k$ .

La figure 3.16 illustre les chemins obtenus, sur une réalisation d'un modèle avec  $V_V = 40\%$ . Plus on augmente  $k$  et moins il y a de chemins parcourus. Le choix de ce paramètre de parcimonie dépend donc de l'application, s'il est nécessaire de faire un compromis entre vitesse et précision.

#### 3.6.3 Comparaison avec d'autres méthodes

On propose dans cette section, de comparer les courbes granulométriques obtenues avec différentes méthodes : le supremum des ouvertures 1-D, les POs, les POs contraints et les PPOs. On utilisera tout d'abord des images provenant de modèles aléatoires puis nous testerons les opérateurs sur une image réelle.

#### Modèles aléatoires : fibres linéaires, fibres tortueuses

L'élément de base des modèles aléatoires sont des fibres linéaires de tailles et d'orientations quelconques. Pour compléter cette étude, nous introduisons un modèle aléatoire utilisant un autre grain

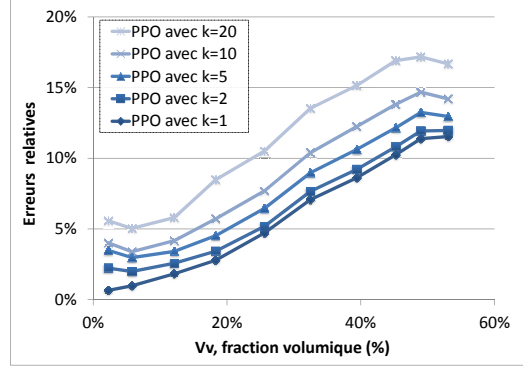


FIGURE 3.15 – Evolution de l’erreur relative entre les courbes granulométriques théorique et expérimentale en fonction de la densité des structures dans l’image et pour différents  $k = \{1, 2, 5, 10, 20\}$ .

primaire : une fibre qui ne sera plus nécessairement droite. Pour générer une fibre tortueuse, nous utilisons un algorithme des marches aléatoires [148] quelque peu modifié ; on ne tolère pas deux changements d’orientation successifs  $+45^\circ$ ,  $-45^\circ$  ou l’inverse. Le pseudo code est présenté dans l’algorithme 6 et les lignes 5 à 10 mettent en œuvre cette contrainte. Les images de la figure 3.17 présentent quelques réalisations de ce processus avec des densités égales à 10%, 20%, 30% et 40%.

---

**Algorithme 6:**  $fibres \leftarrow MarcheAléatoireContraint(l)$

---

**Input :**  $l$  : longueur de la fibre ;

$W, H$  : taille de l’image de sortie ;

**Result :**  $fibres$  : fibre binaire de longueur  $l$

```

1  $lastDir \leftarrow 0$ 
2  $x \leftarrow W/2 - l/2$ 
3  $y \leftarrow H/2$ 
4 for  $rp = x \dots x + l$  do
5   if  $lastDir == 0$  then
6      $dir = rand(-1, 0, 1)$ 
7   else if  $lastDir == -1$  then
8      $dir = rand(-1, 0)$ 
9   else
10     $dir = rand(0, 1)$ 
11     $lastDir = dir$ 
12     $y \leftarrow y + dir$ 
13     $fibres(rp, y - 1) = 255$ 
14     $fibres(rp, y) = 255$ 
15     $fibres(rp, y + 1) = 255$ 

```

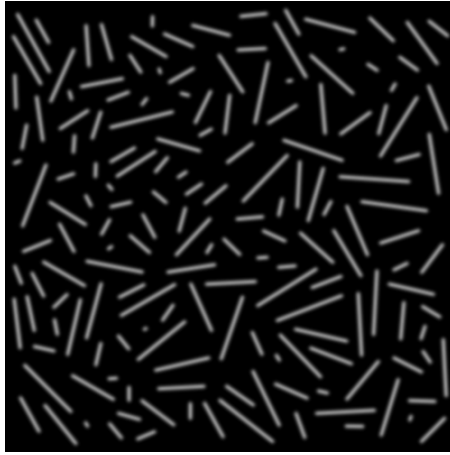
---

On génère 10 réalisations de modèles aléatoires avec des fibres linéaires comme grain primaire pour un premier test puis, avec des fibres tortueuses générées avec un algorithme des marches aléatoires

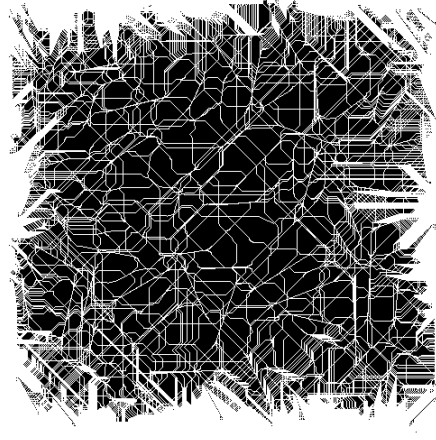


### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

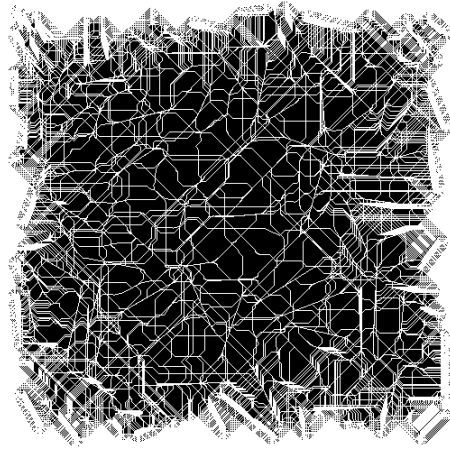
---



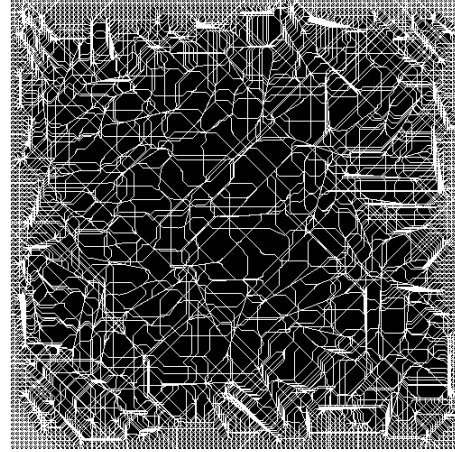
(a) Fraction volumique  $V_V = 40\%$



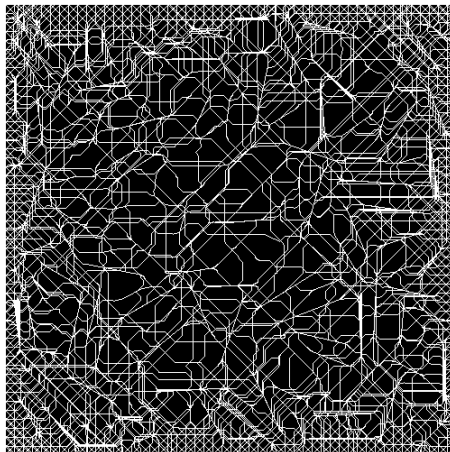
(b) Chemins *loc* avec  $k = 1$



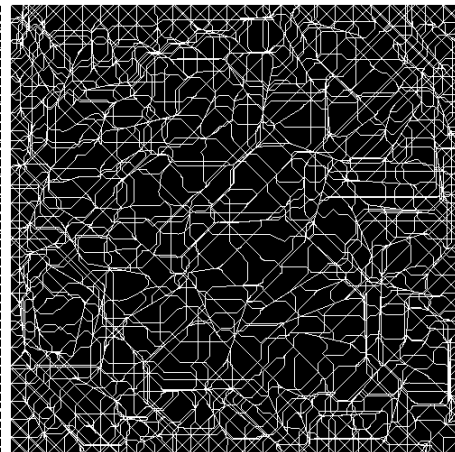
(c) Chemins *loc* avec  $k = 2$



(d) Chemins *loc* avec  $k = 5$



(e) Chemins *loc* avec  $k = 10$



(f) Chemins *loc* avec  $k = 20$

FIGURE 3.16 – Illustration des chemins localement maximaux sur un modèle de fibre avec une fraction volumique de 40%

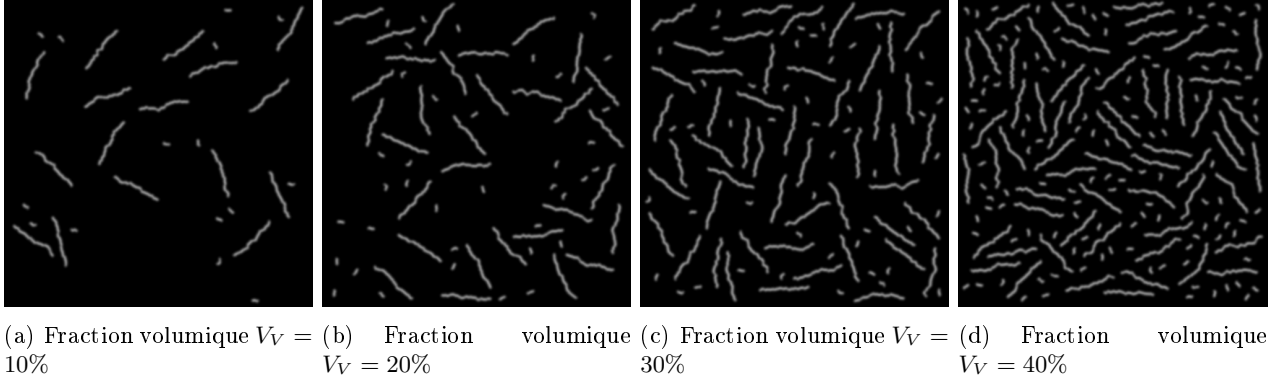


FIGURE 3.17 – 4 réalisations du processus de génération de modèles aléatoires avec des fractions volumiques différentes. Les grains primaires sont des fibres générées avec les marches aléatoires (pseudo code 6) de tailles 10 et 80 pixels

pour le second. On calcule la distribution de taille moyenne associée (pattern spectrum) pour toutes les méthodes décrites : PO, PO contraint, le supremum des ouvertures 1-D, les PPOs avec  $k=1, 2, 5, 10$  et 20. La figure 3.18(a) regroupe l'ensemble des résultats obtenus avec le modèle utilisant une fibre linéaire comme grain primaire (la densité est de 30%, l'orientation est aléatoire et la valeur de l'écart type du filtre gaussien est fixée à 2. La taille des fibres peut prendre aléatoirement 2 valeurs, soit 10 pixels, soit 80 pixels).

**REMARQUE :** La tortuosité des fibres du modèle a été choisie en fonction des capacités de la méthode.

Les ouvertures par chemins classiques surestiment très largement la distribution de taille puisque de nombreuses fibres ont été mesurées avec des longueurs supérieures à 100 pixels. Les PPOs, les POs contraints ou le supremum des ouvertures linéaires ont des *pattern spectra* semblables même si les POs contraints semblent sous-estimer la longueur des plus grandes structures. Lorsque les fibres sont tortueuses (figure 3.18(b)), le supremum des ouvertures par des segments de droite sous-estime largement la taille des structures des images. En effet, il mesure la taille des segments de droite et par conséquent, cet opérateur est inadapté pour mesurer des objets non linéaires. En utilisant des PPOs, les distributions de taille entre les deux modèles sont très stables et on note peu de modifications. Ceci est une bonne chose puisque la tortuosité des fibres est le seul paramètre du modèle qui change et ce paramètre n'a aucune influence sur la distribution de taille.

De plus, on observe la même tendance que dans la section 3.6.2, la taille des fibres est légèrement sous estimée lorsque  $k$  est grand. Néanmoins, on remarque que le paramètre de parcimonie n'a que peu d'influence sur les résultats des PPOs.

**REMARQUE :** D'autres mesures donnent des résultats similaires à ceux des ouvertures parcimonieuses par chemins pour des fibres tortueuses : le diamètre géodésique et des mesures basées sur l'axe médian (voir le chapitre 5).

Pour conclure cette section, l'amélioration la précision des ouvertures parcimonieuses par chemins a rendu cet opérateur invariant par rotation. On a visualisé l'influence du paramètre de parcimonie ainsi que celle des zones aveugles. Ce filtre, bien que dépendant du contenu de l'image, permet de réaliser

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

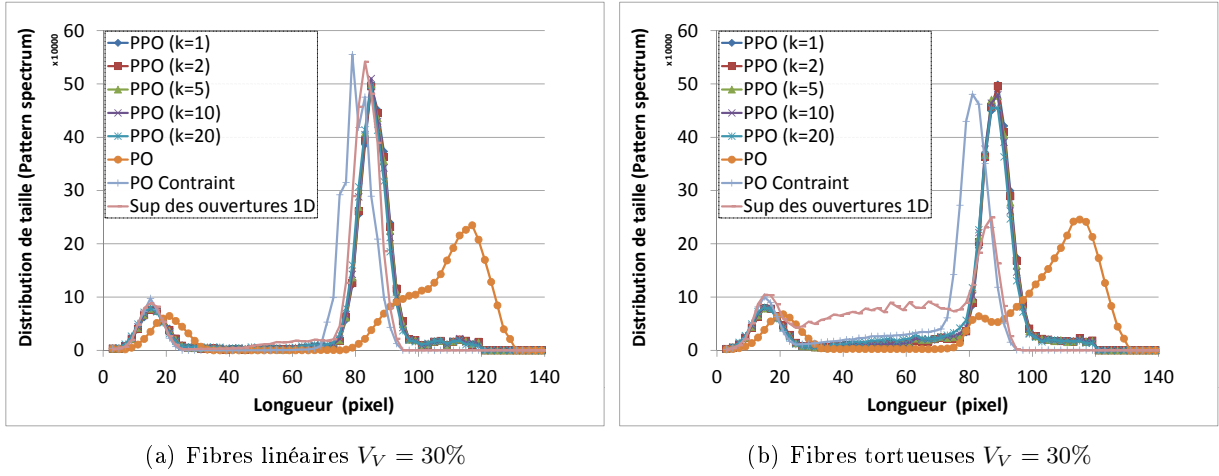


FIGURE 3.18 – Analyse de la distribution de taille de deux modèles aléatoires (moyenne sur 10 réalisations). La taille des fibres est aléatoirement choisie entre deux valeurs 10 pixels ou 80 pixels. On note qu’un filtre gaussien d’écart type 2 est appliqué à ces fibres ce qui augmente leur longueur.

des analyses granulométriques précises des structures. Cet opérateur est même le plus précis de tous ceux testés, pour mesurer des structures filiformes et tortueuses dans une image. La section 3.7 étudie la complexité, les temps de calcul et présente quelques résultats pour des applications industrielles.

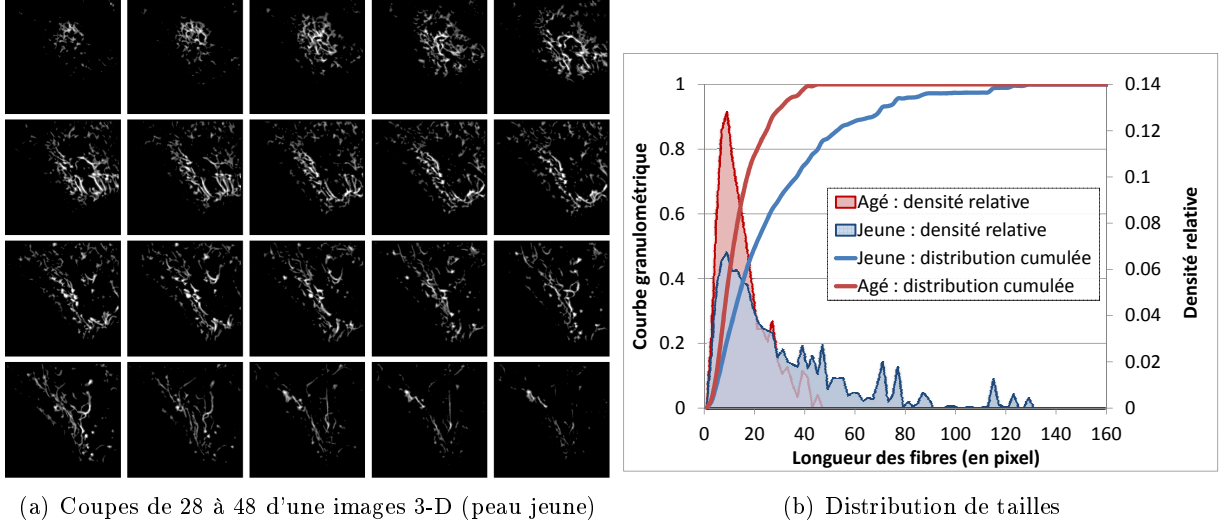
## 3.7 Résultats : applications industrielles

On regroupe dans cette section les résultats de deux applications différentes, l’analyse de la distribution de taille des fibres de la peau, le contrôle de la cornée et la détection des défauts de surface de pièces métalliques (annexe III).

### 3.7.1 Projet L’Oréal : Analyse de la distribution de taille des fibres de la peau

Dans le cadre d’une collaboration avec L’Oréal, nous disposons d’images 3-D de fibres d’élastine du derme de la peau. Ces fibres ont été préalablement rehaussées par un filtre type *Difference of Gaussians* (voir l’étude L’Oréal du chapitre 8), puis un chapeau haut de forme (Top Hat) de rayon 3 est appliqué, suivi de la suppression des valeurs inférieures à un niveau de gris égal à 15 pixels. La figure 3.19 montre quelques exemples de coupes d’une image 3-D.

Nous cherchons donc à calculer la distribution de taille des fibres de la peau pour ainsi déterminer les différences qui existent entre une peau jeune et une peau âgée. Les études que l’on a menées précédemment dans la section 3.6 nous permettent d’avoir plus de recul par rapport aux résultats donnés par les PPOs. On calcule donc des granulométries sur des images 2-D (coupe par coupe dans l’axe  $oxy$ ) avec des ouvertures parcimonieuses par chemins. La figure 3.19 regroupe les courbes granulométriques normalisées ainsi que les densités relatives pour deux types de peau : une peau d’une personne âgée et une peau d’une personne jeune. Des études histologiques [61] ont montré que les fibres d’élastine deviennent plus courtes avec l’âge et on observe cette même tendance sur les courbes granulométriques. Dans le chapitre 8, on étendra cette méthode pour des images 3-D.



(a) Coupes de 28 à 48 d'une images 3-D (peau jeune)

(b) Distribution de tailles

FIGURE 3.19 – Analyse de la distribution de taille des fibres élastines de du derme en fonction de deux types de peau : une peau d’une personne âgée (rouge) et une peau d’une personne jeune (bleu). (La densité relative correspond aux histogrammes, associée à l’axe droit du graphique. La courbe granulométrique correspond aux courbes et à l’axe gauche.)

### 3.7.2 Contrôle de la cornée

On illustre les ouvertures parcimonieuses par chemins en utilisant une image de la cornée [178] obtenue avec un microscope confocal. On cherche ici à détecter les nerfs (long filaments clairs corrompus par du bruit) et à réduire le bruit de l’image. On remarque que les PPOs avec des chemins complets permettent d’obtenir le même niveau de détection que les ouvertures par chemins classiques. De plus, utiliser des ouvertures parcimonieuses pour des chemins incomplets ( $tol = 15$ ) permet de mieux préserver les nerfs de la cornée, même s’ils ont été déconnectés à cause du bruit. Néanmoins, les temps de calcul sont largement à l’avantage des ouvertures parcimonieuses par chemins puisqu’ils sont respectivement égaux à 990 ms, 17 ms et 27 ms pour les POs, les PPOs complets (sans reconstruction) et les PPOs incomplets (sans reconstruction), pour cette image 605 x 504 x 8 bits.

## 3.8 Complexité algorithmique et temps de calcul

On donne dans cette section, la complexité algorithmique de ces opérateurs parcimonieux. Dans un second temps, des benchmarks montrent l’amélioration apportée par les PPOs sur les temps de calcul ainsi que l’influence de  $k$ ,  $tol$  et de la taille de l’image.

### 3.8.1 Complexité algorithmique

Les chemins localement maximaux sont initialisés à partir des bords de l’image (tous les  $k$  pixels). Avec les graphes utilisés (figure 3.1) et la stratégie développée dans la section 3.3.2, on compte au total  $\frac{6(Width+Height)}{k}$  chemins avec  $Width$  et  $Height$ , respectivement la largeur et la hauteur de l’image. Pour connaître la complexité algorithmique des PPOs, il faut connaître le nombre de pixels par chemin, or ce nombre n’est pas une constante. En effet, lorsque l’on utilise un graphe diagonal, le nombre

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

---

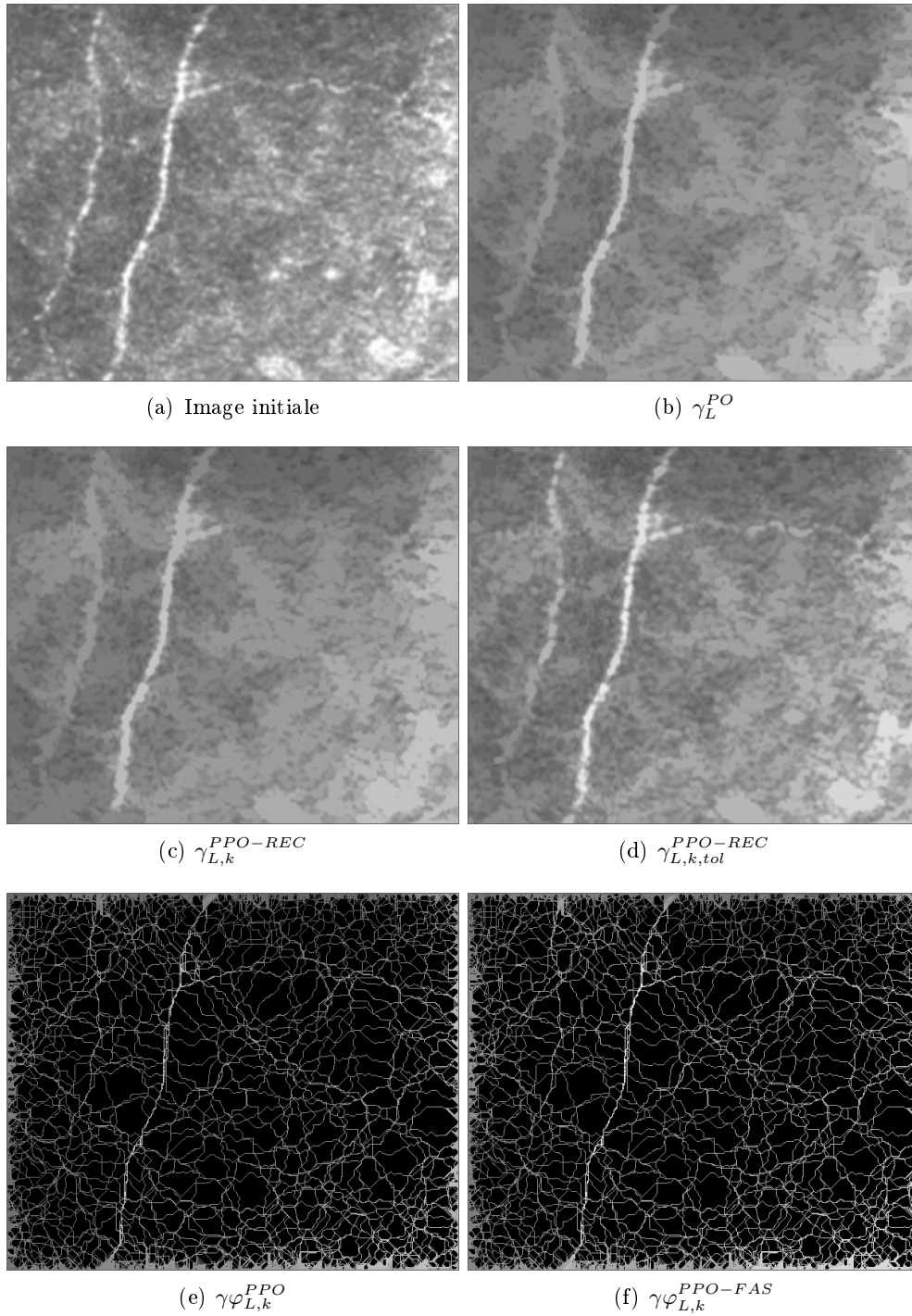


FIGURE 3.20 – Détection nerf de la cornée de l'œil : (a) image initiale, (b) ouverture par chemins de taille  $L = 100$  pixels, (c) ouvertures parcimonieuses par chemins de taille  $L = 100$  pixels avec  $k = 10$ , (d) ouvertures parcimonieuses pour des chemins incomplets  $L = 100$ ,  $k = 10$  avec une tolérance  $tol$  de 15 pixels et (e), (f) un filtre alterné et un filtre alterné séquentiel de taille 100.

de pixels par chemin dépend du contenu de l'image (avec un graphe NO/SE et le premier pixel comme point de départ, le chemin peut être composé de  $Width+Height$  pixels). Au pire des cas, on compte  $\frac{Width(Width+1)}{2}$  pixels analysés pour le bord nord de l'image avec un graphe diagonal. Pour les 4 directions de l'espace, un total de  $\frac{2 \times Width \times Height(Width+Height+4)}{k}$  pixels seront analysés. Néanmoins, le résultat de chaque pixel est calculé avec l'algorithme défini dans le chapitre 2, qui a une complexité constante par pixel. Ainsi, les ouvertures parcimonieuses par chemins ne dépendent pas du paramètre de taille  $L$  et la complexité théorique est égale à  $\mathcal{O}\left(\frac{Width \times Height(Width+Height)}{k}\right)$ . La complexité de cet opérateur est donc quadratique mais si on utilise uniquement les graphes horizontaux et verticaux, alors la complexité devient linéaire en fonction du nombre de pixels de l'image :  $\mathcal{O}\left(\frac{Width \times Height}{k}\right)$ .

Pour les ouvertures parcimonieuses pour des chemins incomplets, il est nécessaire de calculer une fermeture avant de calculer son ouverture sur un signal 1-D. Néanmoins, une fermeture sur un signal 1-D est calculée en utilisant l'équation 3.18 et par conséquent, la complexité reste identique.

On montre expérimentalement à travers les différents benchmarks de la section 3.8.2, que les paramètres de taille  $L$  et  $tol$  n'ont aucune influence sur les temps d'exécution.

#### 3.8.2 Temps de calcul

Les ouvertures parcimonieuses par chemins ont été principalement conçues pour répondre aux contraintes temporelles des POs classiques. Pour illustrer les temps d'exécution et comparer les PPOs avec d'autres opérateurs, on réalise le test suivant : on mesure les temps de calcul pour 100 ouvertures par chemins pour différents  $L$ , pour différents  $k$  et pour différents  $tol$  puis, on compare ces résultats avec les temps de calcul des ouvertures par chemins complets et incomplets ainsi qu'avec les ouvertures par chemins contraints. La figure 3.21 donne l'ensemble des temps moyens pour chacune de ces méthodes en fonction de  $L$ . On note que les temps d'exécution sont reportés sur une échelle logarithmique faisant intervenir 4 ordres de grandeurs. Ce benchmark a été réalisé sur un ordinateur portable, Intel Core 2 Duo T7700 CPU @2.4GHz, sur l'image 8 bits présentée dans la figure 1.1 ( $768 \times 576$  pixels.)

Les temps de calcul constituent l'atout majeur des ouvertures parcimonieuses par chemins. En effet, les PPOs sont indépendants de  $L$  et pour des chemins complets, les temps de calcul ont été divisés par 8, 39 et 77 entre une PO et une PPO avec  $k = \{1, 5, 10\}$  respectivement. L'écart est encore plus grand pour des chemins incomplets, puisque les temps moyens ont été divisés par un facteur proche de 3000.

Un deuxième benchmark est proposé pour visualiser l'impact sur les temps de calcul, du paramètre de parcimonie  $k$  ainsi que de  $tol$ , la tolérance pour des chemins discontinus. On trace donc les temps d'exécution moyens d'une ouverture parcimonieuse par chemins en fonction de  $k$  puis de  $tol$ . Les résultats de ces tests sont reportés dans les figures 3.22(a) et 3.22(b). On note tout d'abord, que les temps de calcul décroissent en  $\frac{1}{k}$  avec le paramètre de parcimonie. En effet, la décroissance des temps est linéaire avec une échelle log/log. Ceci est conforme à nos attentes puisque le nombre de chemins analysés est divisé par  $k$  et que chaque chemin est analysé en temps constant par pixel. Pour le paramètre  $tol$ , son augmentation engendre une diminution des temps de calcul. L'explication de ce phénomène est décrite dans le chapitre 2 et vient du fait que plus la taille de la fermeture est grande et moins il y a de phase d'écritures sur le signal de sortie. Ceci accélère sensiblement l'algorithme.

Enfin un troisième benchmark propose de visualiser l'impact de la taille de l'image sur les temps de calcul. Pour cela, on définit des images de taille croissantes que l'on remplit aléatoirement avec du bruit uniforme. Les résultats des temps d'exécutions moyens d'une PPO sont tracés dans la figure 3.23. Comme défini dans la section traitant de la complexité (section 3.8.1), les temps de calcul sont *quadra-*

### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

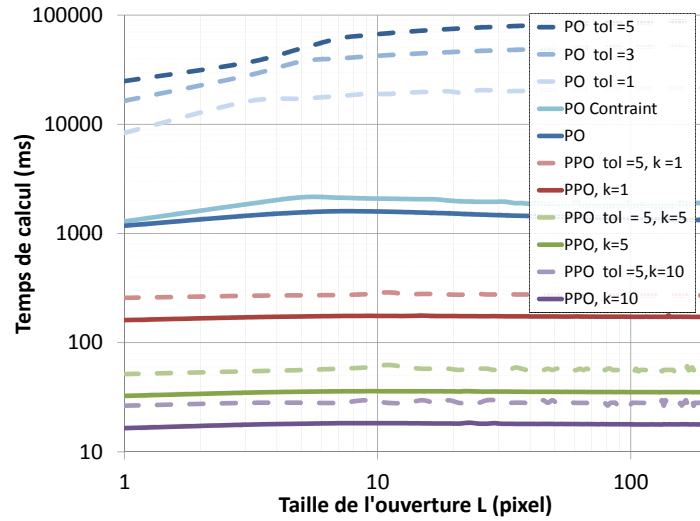


FIGURE 3.21 – Benchmark entre les ouvertures par chemins classiques et les ouvertures parcimonieuses avec des chemins complets ou incomplets. Les temps de calcul sont reportés en ms et sur une échelle logarithmique en fonction de la taille de l'ouverture (taille de l'image  $768 \times 576$  pixels).

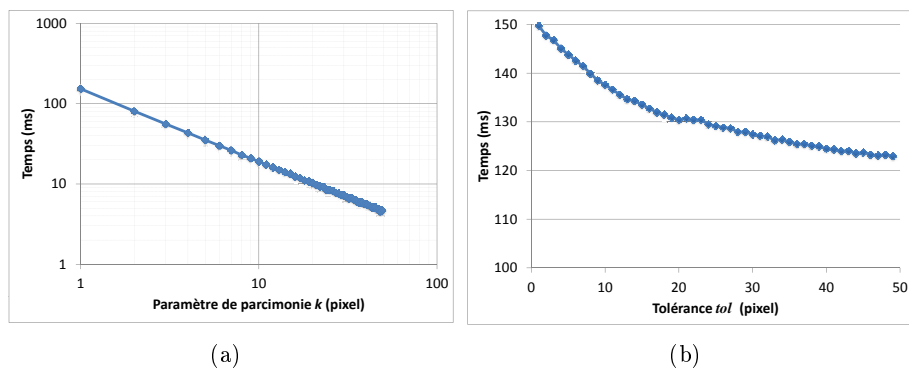


FIGURE 3.22 – Temps de calcul moyen d'une ouverture parcimonieuse par chemins (100 itérations) en fonction de  $k$  et de  $tol$ . L'image test est présentée dans la figure 3.20(a) ( $768 \times 576$  pixels.)

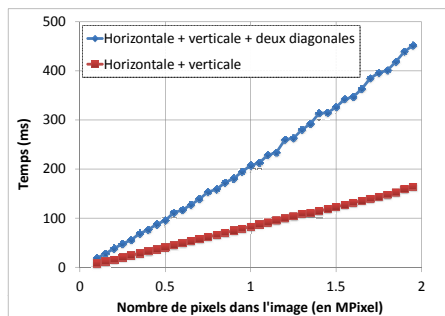


FIGURE 3.23 – Temps de calcul moyen d’une ouverture parcimonieuse par chemins ( $L = 100$ ,  $k = 1$ ,  $tol = 0$ ) en fonction de la taille de l’image (bruit uniforme). On trace en bleu les PPOs avec les 4 directions et en rouge, les PPOs avec seulement 2 directions (horizontale et verticale).

*tiques* si on utilise les graphes diagonaux et *linéaires* si on utilise uniquement les graphes horizontaux et verticaux.

### 3.9 Bilan et perspectives

Ce chapitre présente un nouvel opérateur morphologique : les ouvertures parcimonieuses par chemins. Ces outils sont utilisés pour extraire les structures claires d’une image, qui ont une longueur supérieure à un seuil donné.

Trois stratégies d’extraction d’un sous ensemble de chemins représentatif ont été étudiées et analysées. Nous avons retenu la méthode d’extraction *Loc*, qui construit un chemin localement maximal à partir de tous les pixels du bord de l’image comme point de départ. Cette stratégie présente plusieurs avantages : la taille des régions aveugles est faible, l’extraction des chemins est facile et rapide et chaque chemin est individualisé. Cette dernière caractéristique permet d’utiliser l’algorithme rapide défini dans le chapitre 2, pour calculer une ouverture 1-D sur un chemin. Les PPOs bénéficient alors des avantages de cet algorithme : les temps de calculs sont faibles, des données de toutes précisions peuvent être utilisées et la complexité algorithmique est constante par pixel. La taille de l’ouverture n’a donc plus un impact majeur sur les temps de calcul et la complexité globale est quadratique dans la version avec toutes les orientations et linéaire si on ne sélectionne que les directions verticales et horizontales.

De plus, les PPOs sont étendues aux chemins incomplets. À cause du bruit d’acquisition, des chemins peuvent être discontinus et cet opérateur permet d’insérer une tolérance pour reconnecter ces chemins. Il fournit des résultats similaires aux ouvertures par chemins incomplets en quelques millisecondes.

Les temps de calcul constituent l’atout majeur des ouvertures parcimonieuses par chemins. Les PPOs sont plus rapides de plusieurs ordres de grandeurs par rapport aux ouvertures par chemins classiques et pour des chemins complets et incomplets, les temps de calculs peuvent être respectivement divisés par 77 ou par 3000. Ces opérateurs peuvent donc être utilisées dans des applications où les temps de calcul sont primordiaux, comme le contrôle non destructif de pièces industrielles (chapitre 7).

Cet opérateur n’extraît qu’un sous ensemble de chemins et des erreurs peuvent apparaître dans le résultat du filtrage. Ce chapitre propose de nombreux tests pour étudier et améliorer la précision



### 3. OUVERTURES PARCIMONIEUSES PAR CHEMINS

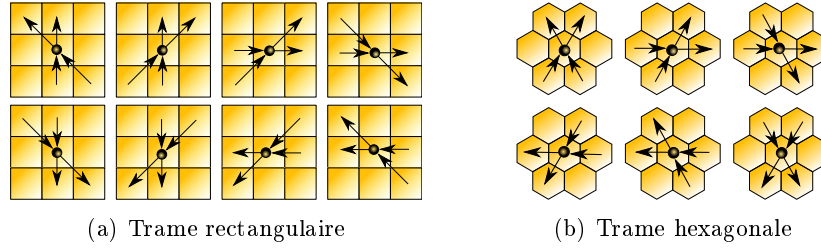


FIGURE 3.24 – Exemple de graphe à 2 voisins pour une trame rectangulaire et une grille hexagonale.

de cet opérateur. Les PPOs sont donc invariants par rotation, une propriété que les ouvertures par chemins classiques ne possèdent pas. Lorsque la densité de l'image augmente tout comme le paramètre de parcimonie, les erreurs commises par les PPOs augmentent. Néanmoins, les PPOs sont plus précis pour estimer la distribution de taille d'une image, car les ouvertures par chemins classiques surestiment fortement la taille des structures et les ouvertures par chemins contraints, les sous-estiment.

On pourrait utiliser d'autres graphes pour construire les ouvertures parcimonieuses par chemins. On s'est intéressé à des graphes ayant seulement deux successeurs et deux prédécesseurs pour construire aussi bien des PPOs que des POs. Ces graphes sont présentés dans la figure 3.24 pour une trame rectangulaire et une trame hexagonale. Ils ne surestiment pas les chemins diagonaux et sont invariants par rotation. Cependant, même s'ils paraissent séduisant en théorie, la détection des structures tortueuses est beaucoup moins bonne. En effet, les fibres se retrouvent souvent morcelées avec ce type de graphe et donc, leur longueur réelle est sous-estimée. On peut donc définir une sorte de hiérarchie des opérateurs à utiliser en fonction des structures à détecter :

- Pour des structures droites, il est préférable d'utiliser le supremum des ouvertures linéaires (chapitre 2).
- Pour des structures très faiblement tortueuses, on utilisera des PPOs avec un graphe à 2 successeurs sur une trame hexagonale.
- Pour des structures raisonnablement tortueuses, on utilisera des PPOs avec un graphe à 3 successeurs avec une trame rectangulaire.
- Pour des structures fortement tortueuses, l'algorithme développé dans le chapitre 5 sera utilisé puisque les ouvertures par chemins couperont les structures tortueuses en plusieurs morceaux.

Pour étendre l'utilisation des opérateurs parcimonieux, nous pouvons définir d'autres filtres comme les "Difference Of Gaussian" parcimonieux. Les filtres DoG, introduits dans le chapitre 1 nécessitent d'utiliser une banque de filtres directionnels. Utiliser des chemins localement maximaux permet d'avoir un filtre parcimonieux ayant une complexité beaucoup moins importante. En effet, l'orientation du chemin est connue, il n'est donc pas nécessaire de calculer le résultat du filtre pour chaque orientation. Les temps de calcul seront donc fortement réduits.

L'extraction des chemins est pour le moment réalisée en utilisant une recherche locale du prochain pixel à insérer au chemin en cours de construction. Cette recherche locale peut rendre cet opérateur sensible au bruit et cette sensibilité peut être améliorée en faisant des recherches moins locales. On définira des chemins *semi optimaux* qui seront une généralisation des chemins localement et globalement optimaux, permettant de dicter un comportement intermédiaire entre ces deux méthodes d'extraction de chemins. Les PPOs seront donc plus robustes au bruit mais plus lent.

---

## Éléments structurants adaptatifs : nouveaux éléments structurants adaptatifs basés sur un processus de croissance de région et nouveaux opérateurs qui analysent leurs formes

---

Les ouvertures parcimonieuses par chemins, développées dans le chapitre précédent (chapitre 3), permettent de filtrer les structures de l'image plus petites qu'un seuil donné. Néanmoins, le graphe sous-jacent à l'image a une influence, fixant l'ensemble des structures que ce filtre est capable de détecter. Pour des fibres tortueuses ne contenant aucun chemin du graphe, les ouvertures par chemins sous-estiment leur taille réelle. Dans ce contexte et pour supprimer cette contrainte, on propose dans ce chapitre et le suivant (chapitre 5), deux nouveaux opérateurs permettant d'extraire des structures fines ayant une tortuosité quelconque.

Ce chapitre introduit des nouveaux éléments structurants adaptatifs, qui ont une taille fixe mais une forme variable. Ainsi, la forme de l'élément structurant s'adapte au contenu de l'image en choisissant récursivement, les pixels les plus proches en intensité à l'origine de l'ES. On appelle cette méthode, *region growing structuring element* (REGSE), que l'on peut traduire en français comme *élément structurant par croissance de région*.

Avec les REGSEs, on peut non seulement calculer les opérateurs morphologiques classiques utilisant un élément structurant, ainsi que des opérateurs linéaires utilisant un noyau, mais en plus, on introduit une méthode originale qui analyse la forme des éléments structurants adaptatifs pour obtenir des informations complémentaires sur le contenu de l'image. On obtient un nouvel ensemble d'opérateurs, capables de rehausser efficacement les structures fines de l'image.

On illustre les performances des nouveaux opérateurs proposés en prenant des images associées aux projets TOCATA (chapitre 7) et L'Oréal (chapitre 8).

On comparera cette méthode avec d'autres opérateurs adaptatifs, comme les amibes morphologiques et les *general adaptive neighborhood structuring elements* (GANSE). On constatera que les opérateurs basés sur les REGSEs proposent une meilleure détection pour nos applications.

Mots clés : Morphologie mathématique, filtrage, réduction du bruit, rehaussement, détection des structures fines, region growing structuring element, inertie, élongation, excentricité.

Ce travail a donné lieu à une publication :

1. Dans Morard *et al.* [151], les nouveaux éléments structurants sont introduits ainsi que la description des opérateurs utilisant la forme des REGSEs pour obtenir de l'information sur le contenu de l'image.

**V. Morard, E. Decenci re, and P. Dokl dal, Region Growing Structuring Elements and New Operators based on their Shape, *International conference on Signal and Image Processing (SIP)*, Dallas, USA, 2011**

### 4.1  tat de l'art

R aliser un filtrage adaptatif permet de mieux prendre en compte le contenu de l'image. L'objectif principal consiste g n ralement,   filtrer les r gions homog nes tout en conservant les r gions de fort gradient. De nombreux op rateurs adaptatifs existent dans la litt rature et ils sont divis s en deux cat gories principales : les filtres lin aires et les filtres non lin aires.

Pour les filtres lin aires, Kuwahara *et al.* [113] ainsi que Nagao [161] ont divis  le noyau en quatre r gions autour du pixel central. Apr s avoir calcul  l' cart type pour chaque r gion, le pixel central prendra la valeur moyenne de la r gion la plus homog ne. Ce filtre permet de pr server les contours tout comme d'autres filtres adaptatifs tels que la diffusion anisotrope de Perona et Malik [183] ou le filtre bilat ral de Tomasi *et al.* [234]. Le filtre bilat ral est un filtre dont le noyau est pond r  non seulement par une distance spatiale mais aussi par une distance tonale. Ce filtre, utilis  dans de nombreuses applications [157, 158, 184], pr serve les contours de l'image. Enfin, le filtre *mean shift* [41] utilise des m thodes de classification pour prendre la valeur moyenne d'une r gion homog ne.

Pour les filtres morphologiques (op rateurs non lin aires), remplacer les  l ments structurants fixes par des  l ments structurants variant dans l'espace, constitue une solution  l gante pour prendre en compte plus finement le contenu de l'image. Ces ESs sont appel s des fonctions structurantes par Serra [203] et cette adaptabilit  peut  tre r alis e de plusieurs fa ons diff rentes. Dans une application de contr le routier, Beucher *et al.* [20] ajust rent la taille de l'ES   sa position dans l'image pour corriger l'effet de la perspective. La taille des ESs augmente donc lin airement avec la position verticale dans l'image du v hicule consid r . La forme de l' l ment structurant peut aussi  tre ajust e avec le contenu de l'image. Ainsi, Gordon et Rangayyan [78], introduisirent la notion de *voisinage adaptatif*. La taille de l'ES est choisie pour maximiser une fonction de contraste. Cela a amen  Braga-Neto [26], puis Debayle et Pinoli [48],   introduire la notion de *General Adaptive Neighborhood Structuring Elements* (GANSE). Les  l ments structurants sont d finis en termes de connectivit  d'ordre  $m$  et n'ont pas de contraintes de taille. Une autre approche a  t  suivie par Lerallut *et al.* [118, 119]. Ils propos rent les *amibes morphologiques*, qui sont des ESs dont la forme est pilot e par une distance g od sique sp cifique. Grazzini et Soille repr rent ce principe et introduisirent la notion de temps g od sique [81]. Dans [227], Tankyevych *et al.* ont travaill  sur des  l ments structurants rectilignes dont l'orientation s'adapte   l'orientation locale des structures fines   d tecter. D'autres approches ont  t  introduites par Salembier [194, 196], Chen et Venetsanopoulos [37] et d'autres auteurs [149, 45, 16, 242]. Tous les aspects th oriques des  l ments structurants variant dans l'espace ont  t   tudi s par Bouaynaya *et al.* pour des images binaires [23] et pour des images en niveaux de gris [24].

  partir des diff rentes approches propos es dans la litt rature, les amibes morphologiques apparaissent comme une solution prometteuse. Les amibes tendent   avoir une forme circulaire mais elles

peuvent s'adapter au contenu de l'image. Leurs formes sont guidées par une distance géodésique généralisée. Soit  $\pi = \{x_0, \dots, x_i, \dots, x_n\}$  un chemin entre deux points  $x_0$  et  $x_n$ . La longueur de ce chemin est définie par  $L(\pi) = \sum_{1 \leq i \leq n} (1 + \lambda d_{\text{pixel}}(x_{i-1}, x_i))$ , où  $d_{\text{pixel}}$  est la différence absolue en niveaux de gris entre deux pixels et  $\lambda$  un paramètre tempérant l'influence des gradients. Cela permet d'introduire la distance morphologique des amibes entre deux points  $x$  et  $y$  :

$$d_\lambda(x, y) = \min_{\pi \in P(x, y)} L(\pi), \quad (4.1)$$

où  $P(x, y)$  est l'ensemble des chemins reliant  $x$  à  $y$ . Pour la détection des éléments fins, il est nécessaire d'aller plus loin, en supprimant toute contrainte de forme ou de taille aux éléments structurants pour s'adapter pleinement au contenu de l'image. Debayle et Pinoli ont suivi cette idée et proposèrent des éléments structurants sans aucune contrainte. Le seul paramètre de l'opérateur,  $m \in \mathbb{Z}^+$ , contrôle l'homogénéité de l'ES. Soit  $f : D \rightarrow V$  une image et  $x$  le pixel central,  $x_i$  est connecté à  $x$  avec un chemin autorisé  $\pi$  ( $x_i$  appartient à l'ES de  $x$ ) si et seulement si  $\forall x_i \in \pi, |f(x_i) - f(x)| \leq m$ . Ainsi, l'ES est une composante connexe d'ordre  $m$ , composée de pixels  $x_i$  reliés par un chemin autorisé à  $x$ . La complexité de cet algorithme dépend du paramètre  $m$  ainsi que du contenu de l'image. Elle peut être arbitrairement grande puisque les ESs peuvent potentiellement se propager sur toute l'image. Au contraire, si le pixel central se situe sur un pixel affecté par un bruit poivre et sel, l'élément structurant ne pourra pas grandir et le pixel bruité sera conservé.

Pour obtenir un opérateur efficace aussi bien pour la réduction du bruit que pour la détection des structures fines des images, il faut prendre en compte les avantages de chaque méthode. Les nouveaux éléments structurants, appelés REGSE, auront donc une forme quelconque mais avec une contrainte de taille. Si l'ES est positionné sur une structure filiforme très tortueuse de l'image, alors il pourra la suivre et la préserver. De plus, l'analyse de la forme de l'élément structurant permettra de discriminer les structures filiformes des autres. À notre connaissance, c'est la première fois que la forme d'un élément structurant adaptatif est analysée pour extraire de l'information sur le contenu de l'image.

On propose tout d'abord d'introduire dans la section 4.2, le concept, les définitions et les propriétés des REGSEs. Puis la section 4.3 présente le nouvel ensemble d'opérateurs qui analysent la forme de l'ES en utilisant les REGSEs, puis d'autres éléments structurants adaptatifs. Enfin, les sections 4.4 et 4.5 présentent quelques résultats ainsi que les aspects pratiques (pseudo code et optimisation de l'algorithme, la complexité et les temps de calcul).

## 4.2 Region growing structuring elements

Cette section introduit la définition de ces éléments structurants qui seront utilisés pour calculer les opérateurs classiques de la morphologie mathématique.

### 4.2.1 Définition

Soit  $f$  une image en niveaux de gris, une application de  $D$  dans  $V$ .  $D \subset \mathbb{Z}^2$  est le domaine (rectangulaire) de l'image et  $V$  l'ensemble des valeurs,  $V = [0, \dots, M - 1]$ .

#### DÉFINITION 4.1

Pour tout  $x \in D$ ,  $\mathcal{A}_x$  est l'ensemble des éléments structurants tels que  $A_x \in \mathcal{A}_x$  si et seulement si :

$$\begin{aligned} A_x &\subseteq D, \\ x &\in A_x, \\ A_x &\text{ est connexe.} \end{aligned} \quad (4.2)$$

On note  $A_x^L$ , un élément structurant appartenant à  $\mathcal{A}_x$ , qui a exactement  $L$  pixels ( $\text{Card}(A_x^L) = L$ ). La construction de cet élément structurant est réalisée récursivement et utilise un processus de croissance de région d'où l'opérateur tire son nom.  $x$ , est le point initial ou *l'origine de l'ES*, il correspond au germe du processus de croissance de région et  $\text{Card}(A_x^L) = L$  est le critère d'arrêt. Il reste donc à définir le critère d'agrégation d'un pixel à l'ES et les équations 4.3, 4.4 et 4.5 montrent les différentes étapes pour y parvenir :

$$\begin{aligned} A_x^1 &= \{x\} \\ A_x^2 &= A_x^1 \cup \{y_2\} \\ &\vdots \\ A_x^L &= A_x^{L-1} \cup \{y_L\}, \end{aligned} \tag{4.3}$$

où  $y_i$  est un des voisins de  $A_x^{i-1}$ , sélectionné en minimisant une fonction d'homogénéisation,  $\rho$ , qui est décrite par l'équation suivante :

$$\forall a, b \in D, \rho(a, b) = |f(a) - f(b)|. \tag{4.4}$$

Enfin, muni de cette fonction d'homogénéisation,  $y_i$  est un pixel sélectionné parmi l'ensemble des pixels voisins de  $A_x^{i-1}$  (noté  $\mathcal{N}(A_x^{i-1})$ ) avec une 8 connexité écrit  $\mathcal{N}(x) = \{y, \|x - y\|_{\mathcal{L}_\infty}\}$  et qui minimise  $\rho$  :

$$Y = \underset{y_j \in \mathcal{N}(A_x^{i-1})}{\operatorname{argmin}} \rho(x, y_j), \tag{4.5}$$

et qui minimise aussi la distance par rapport au pixel initial de l'élément structurant  $x$  :

$$y_i \in \underset{y_j \in Y}{\operatorname{argmin}} \|x, y_j\|_{\mathcal{L}}. \tag{4.6}$$

$\mathcal{L}_\infty$  correspond à la norme infinie et  $\mathcal{L}$  à une des normes. La figure 4.1 présente quatre éléments structurants adaptatifs REGSEs sur une image test. La position de l'origine de chaque ES est représentée par une croix rouge et les pixels appartenant à cet ES apparaissent en blanc. On observe une grande variabilité dans la forme des REGSEs, hormis la taille de l'ES. En effet, il n'y a pas de contrainte sur la distance d'un pixel au point initial de l'ES du moment que la fonction d'homogénéisation donne un résultat plus faible. Des éléments structurants peuvent donc s'étendre sur l'image (l'ES numéro 3 suit une ligne de niveau du signal initial). On voit ici l'avantage principal des REGSEs, puisqu'ils sont capables de suivre des structures fines, même si ces dernières ne mesurent qu'un ou deux pixels de large. On remarque aussi que sur des zones plates, la forme des éléments structurants est un carré (ES numéro 1). Cette forme est dictée par le choix de la norme dans l'équation 4.6 et l'utilisation des normes  $\mathcal{L}_1$ ,  $\mathcal{L}_2$  ou  $\mathcal{L}_\infty$  donne respectivement un losange, un cercle ou un carré.

Dans le reste de ce chapitre, on utilise exclusivement la norme  $\mathcal{L}_\infty$  dans l'équation 4.6. On note que la contrainte de distance, introduite par l'équation 4.6, permet d'obtenir une forme régulière sur des zones plates. Cette propriété importante sera exploitée dans la section 4.3, lorsque l'on analysera la forme des éléments structurants adaptatifs.

### 4.2.2 Morphologie mathématique

Les opérateurs de base de la morphologie mathématique utilisent ces éléments structurants adaptatifs. En prenant, pour chaque pixel de l'image  $f$ , la valeur minimale de tous les pixels de l'ES, on obtient l'érosion de  $f$  :

$$\epsilon^L f(x) = \bigwedge_{x_i \in A_x^L} f(x_i). \tag{4.7}$$

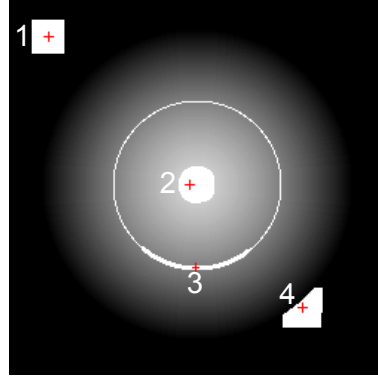


FIGURE 4.1 – Visualisation de la forme de quatre REGSEs avec  $L = 506$  pixels. Les pixels centraux sont représentés par une croix rouge.

La dilatation adjointe est donnée par :

$$\delta^L f(x) = \bigvee_{x_i | x \in A_{x_i}^L} f(x_i). \quad (4.8)$$

Utiliser l'adjonction permet de composer ces deux opérateurs de base pour former des ouvertures et des fermetures, respectivement notées  $\gamma^L$  et  $\varphi^L$ . Le pseudo code 7 présente algorithmiquement, le fonctionnement des ouvertures avec des éléments structurants variant dans l'espace. On note que la forme de l'élément structurant, qui est calculé avec la fonction *ComputeREGSE* (algorithme 9 décrit dans la section 4.5 de ce chapitre), utilise la même image,  $f$ , pour l'érosion et la dilatation adjointe.

---

**Algorithme 7:** Ouverture avec des REGSE :  $f_{out} = \gamma^L(f)$

---

**Input :**  $f$  : image initiale

$L$  : nombre de pixels de l'élément structurant REGSE

**Result :**  $f_{out} = \gamma^L(f)$

```

1 for  $x \in D$  do
    /* Calcul de la forme de l'ES (algorithme 9)                                     */
2      $A_x^L \leftarrow \text{ComputeREGSE}(f, x, L)$ 
3      $f_{out}(x) \leftarrow \min(A_x^L)$ 
4 for  $x \in D$  do
    /* Calcul de la forme de l'ES avec la même image  $f$                              */
5      $A_x^L \leftarrow \text{ComputeREGSE}(f, x, L)$ 
6     for  $y \in A_x^L$  do
7          $f_{out}(y) \leftarrow \max(f_{out}(x), f_{out}(y))$ 
    
```

---

Les éléments structurants par croissance de région adaptent leurs formes aux structures fines et très tortueuses de l'image. C'est un avantage cependant, de telles structures peuvent aussi être trouvées dans le fond de l'image à cause du bruit. Lerallut *et al.* [117] proposèrent d'utiliser une *image pilote* pour calculer la forme des amibes. La plus simple image pilote est l'image initiale mais, pour limiter

l'influence du bruit, on peut utiliser une image simplifiée par un filtre gaussien ou un filtre bilatéral. Cette image pilote intervient uniquement dans le calcul de la forme du REGSE et il suffit de remplacer dans l'algorithme 7, l'image  $f$  (en rouge) par une autre image. L'utilisation d'une image pilote améliore les capacités de réduction du bruit de ce filtre.

**REMARQUE IMPORTANTE:** En utilisant une image pilote avec des éléments structurants adaptatifs, appeler ces opérateurs ouvertures ou fermetures est incorrecte d'un point de vue formel. En effet, la propriété de croissance peut ne pas être respectée. On rappelle que pour qu'un opérateur  $\Psi$  soit croissant, on doit avoir la relation suivante pour tous les pixels du domaine et pour deux images  $f$  et  $g$  :

$$f \leq g \Leftrightarrow \Psi(f) \leq \Psi(g). \quad (4.9)$$

Cependant, cette relation peut ne pas être respectée si on prend des images pilotes quelconques comme le montre la figure 4.2. Les deux signaux  $f$  et  $g$  avec  $f \leq g$  associés aux images pilotes de la figure 4.2(b). Les ouvertures avec des REGSEs de taille  $L = 3$  implique que l'ouverture de  $f$  par des REGSEs de taille 3 n'est pas inférieure à l'ouverture de  $g$ . À notre connaissance, c'est la première fois que ce point est relevé pour la morphologie adaptative or cette propriété affecte tous les éléments structurants adaptatifs. Par abus de langage, on appellera tout de même ces opérateurs des ouvertures et des fermetures, même si l'équation 4.9 n'est pas vérifiée.

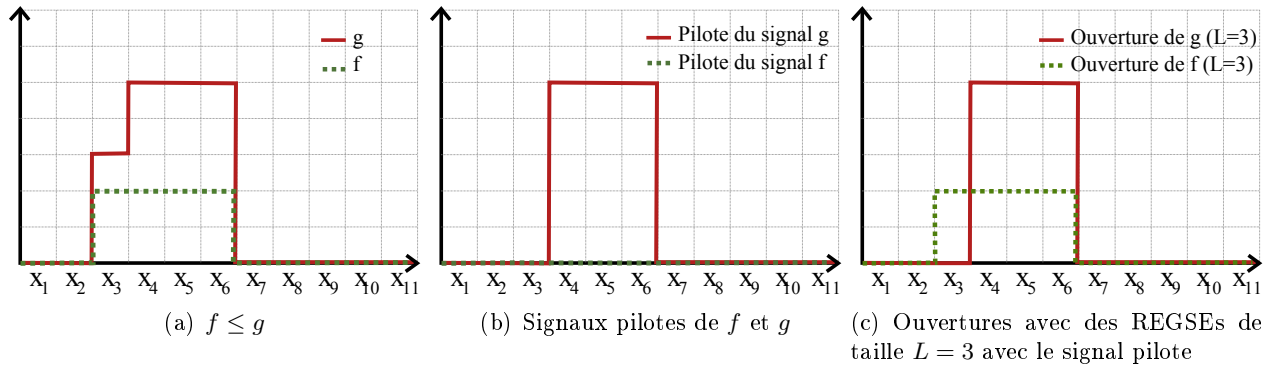


FIGURE 4.2 – Les opérateurs associés aux éléments structurants adaptatifs ne sont pas croissants. Pour les signaux  $f$  et  $g$  de (a), on a  $f \leq g$ . Or en utilisant des signaux pilotes quelconques pour  $f$  et pour  $g$  (b), le résultats des ouvertures avec des REGSEs de taille  $L = 3$  ne donne pas  $\gamma^3(f) \leq \gamma^3(g)$  pour tous les pixels (cf pixel  $x_3$ ).

Les résultats d'une ouverture et d'une fermeture avec des éléments structurants adaptatifs REGSEs sont présentés dans les figures 4.6(b) et 4.6(c). L'image pilote utilisée est l'image initiale filtrée par une gaussienne d'écart type 1. Les REGSEs s'étendent prioritairement sur les régions homogènes par rapport à l'origine de l'ES et ces deux filtres préservent donc les frontières des objets, sauf lorsque la structure analysée est plus petite que la taille de l'ES. En ce sens, le comportement des REGSEs se rapproche des ouvertures et fermetures aréolaires [243].

### 4.2.3 Propriétés des REGSEs

Il est intéressant de noter que pour des images **binaires**, les ouvertures par REGSEs donnent exactement les mêmes résultats que des ouvertures aréolaires. En effet, les REGSEs ont une taille fixe et toutes les structures ayant une taille inférieure à ce paramètre seront supprimées lors de la phase d'érosion. Par dualité, les fermetures REGSEs sont équivalentes aux fermetures aréolaires.

#### Suppression des marches lors des transitions entre deux régions

L'équivalence entre ces deux opérateurs n'est plus vraie pour des images en niveaux de gris. La figure 4.3 présente un exemple de situation où les deux opérateurs donnent des résultats différents. Pour ce signal 1-D, une ouverture aréolaire de taille 3 ne modifie pas le signal initial, alors qu'une ouverture avec des REGSEs de même taille permet de filtrer la petite marche du signal (pixel  $x_6$ ). Par construction, l'élément structurant adaptatif du pixel  $x_6$  est  $A_{x_6}^3 = \{x_6, x_7, x_8\}$ . Lors de la phase d'érosion, ce pixel est supprimé puisqu'il prend la valeur du pixel  $x_7$  (c'est le seul changement lors de cette phase). Enfin, la dilatation adjointe ne modifie pas la valeur du pixel  $x_6$  puisqu'il ne fait partie d'aucun autre REGSE. Avec cet exemple, on montre qu'utiliser cet opérateur avec de petites valeurs pour  $L$ , améliore le contraste de l'image en supprimant les marches lors des transitions entre deux régions.

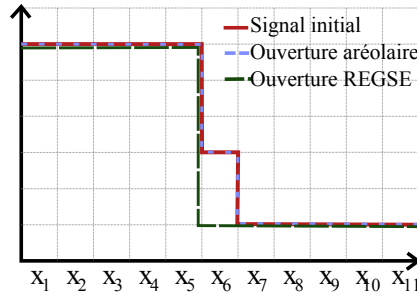


FIGURE 4.3 – Signal 1-D filtré par une ouverture aréolaire et une ouverture par REGSE,  $L = 3$  pixels.

#### Opérateur non connexe

À première vue, les ouvertures et les fermetures basées sur les REGSEs semblent être des opérateurs connexes<sup>1</sup>. Néanmoins, la figure 4.4 présente un contre exemple simple où une nouvelle frontière est générée lors d'une ouverture avec des REGSEs de taille 4. Les pixels  $x_1, x_2$  et  $x_3$  ont le même élément structurant adaptatif :  $A_{x_1}^4 = A_{x_2}^4 = A_{x_3}^4 = \{x_1, x_2, x_3, x_4\}$ , ainsi que les pixels  $x_6, x_7$  et  $x_8$  :  $A_{x_6}^4 = A_{x_7}^4 = A_{x_8}^4 = \{x_5, x_6, x_7, x_8\}$ . Lors de la phase d'érosion, les pixels  $x_1, x_2$  et  $x_3$  sont abaissés à la valeur du pixel  $x_4$  tandis que les pixels  $x_4$  et  $x_5$  sont abaissés d'un cran. Pendant la dilatation adjointe, la zone plate formée par les pixels  $x_1$  à  $x_3$ , relève le pixel  $x_4$  créant ainsi une nouvelle frontière.

1. Un opérateur connexe est un opérateur qui ne crée pas de nouveaux contours dans l'image.



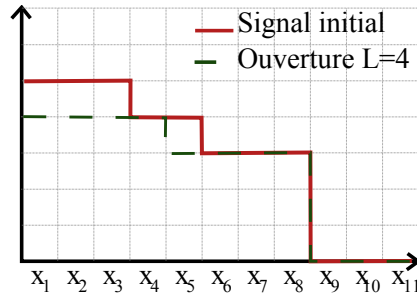


FIGURE 4.4 – Signal 1-D filtré par une ouverture par REGSE de taille 4 pixels

### La famille d'ouvertures REGSEs ne forme pas une granulométrie

On introduit une autre propriété, illustrée dans la figure 4.5. Une ouverture par REGSE de taille 6 peut être localement inférieure à une ouverture de taille 5. En effet, la zone plate formée par les pixels  $x_1$  à  $x_5$  possède le pixel  $x_6$  dans leurs éléments structuraux adaptatifs. Lors de la phase d'érosion, cette zone plate est abaissée à la valeur de  $x_6$ . Cette propriété non intuitive signifie que la famille d'ouvertures avec des REGSEs ne forme pas une granulométrie, comme l'a défini Matheron dans [137].

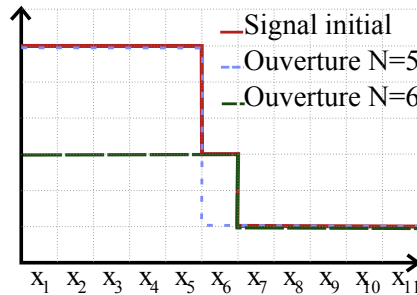


FIGURE 4.5 – Signal 1-D filtré par des ouvertures REGSEs de tailles 5 et 6 pixels

### Invariance par rotation

Les opérateurs basés sur les REGSEs ne sont pas invariants par rotation. En effet, la construction d'un REGSE dépend d'un processus de croissance de régions qui sera stoppé lorsque  $L$  pixels seront insérés dans  $A^L$ . Ainsi, si les équations 4.5 et 4.6 ne permettent pas d'avoir un unique résultat, alors la forme du REGSE peut varier en fonction du sens de parcours de tous les pixels voisins de l'ES.

Un élément structurant basé sur la méthode REGSE permet de suivre des structures filiformes de l'image même si leur largeur est très faible. Cependant, la forme de l'élément structurant sera un carré (avec la norme  $\mathcal{L}_\infty$  dans l'équation 4.6) sur les zones plates de l'image, préservant ainsi sa régularité. Analyser la forme des éléments structuraux variant dans l'espace permet d'obtenir de l'information sur les structures sous-jacentes de l'image et la section 4.3 décrit les nouveaux opérateurs obtenus à partir de cette méthode.



FIGURE 4.6 – Utilisation des REGSEs pour la réduction du bruit sur l'image caméraman. L'image pilote est l'image initiale filtrée par une gaussienne d'écart type 1.

### 4.3 Analyse de la forme des REGSEs

Un nouvel ensemble d'opérateurs peut être créé à partir de l'analyse de la forme des REGSEs. En effet, les éléments structurants adaptent leurs formes aux structures sous-jacentes contenues dans l'image et par conséquent, des informations supplémentaires peuvent être extraites de cette analyse. Notre objectif principal est de distinguer les structures fines des autres et nous développons un opérateur qui utilise l'*inertie*.

#### 4.3.1 Opérateur utilisant le moment d'inertie

Si un REGSE se trouve sur une zone plate d'une image, sa forme sera proche de celle d'un carré. Au contraire, si le pixel initial du REGSE est sur une structure fine et longue, sa forme sera allongée. Pour quantifier ce changement d'aspect, on calcule pour chaque REGSE son moment d'inertie. On

## 4. REGSE

---

introduit alors l'opérateur  $\text{Ine}_{REGSE}^L$ , que l'on définit de la manière suivante,  $\forall x \in D$  :

$$\text{Ine}_{REGSE}^L(x) = \frac{1}{L} \sum_{x_i \in A_x^L} \|x_i - \bar{x}\|_{\mathcal{L}_2}^2, \quad (4.10)$$

avec  $\bar{x} = \frac{1}{L} \sum_{x_i \in A_x^L} x_i$ , le barycentre de l'élément structurant et  $\|x_i - \bar{x}\|_{\mathcal{L}_2}$  la distance euclidienne du point  $x_i$  au barycentre. Plus l'élément structurant est allongée, plus son moment d'inertie sera important et plus la réponse de cet opérateur sera importante, permettant de rehausser toutes les structures fines de l'image. On peut décliner cet opérateur en remplaçant la norme  $\mathcal{L}_2$  par d'autres normes pour obtenir des résultats différents.

Les résultats de cet opérateur sont expliqués et illustrés dans la section 4.4.

### 4.3.2 Opérateur d'extension maximale

Un autre opérateur, ayant un comportement similaire au filtre précédemment proposé, peut être construit. Au lieu de calculer le moment d'inertie de l'élément structurant, on mesure la distance entre le barycentre du REGSE et son point le plus éloigné. De fait, plus la forme de l'ES est allongée et plus l'extension maximale est importante, ce qui permet de définir l'opérateur  $\text{Ext}_{REGSE}^L$ ,  $\forall x \in D$  :

$$\text{Ext}_{REGSE}^L(x) = \max_{x_i \in A_x^L} \|x_i - \bar{x}\|_{\mathcal{L}_2}. \quad (4.11)$$

### 4.3.3 Opérateur d'excentricité

En comparant la distance entre le barycentre et l'origine du REGSE, on peut réaliser une mesure d'excentricité notée  $\text{Exc}_{REGSE}^L$ . Les ESs, dont le pixel initial se situent à une extrémité d'une fibre se retrouvent rehaussés. Cet opérateur permet donc de détecter les extrémités des structures filiformes. :

$$\text{Exc}_{REGSE}^L(x) = \|x - \bar{x}\|_{\mathcal{L}_2}, \quad (4.12)$$

$x$  dénotant l'origine de l'élément structurant et  $\bar{x}$ , son barycentre. Un exemple d'utilisation de cet opérateur est donné dans la section 4.4.

### 4.3.4 Rehaussement des structures claires, des structures sombres

Tous ces opérateurs analysent la forme des éléments structurants sans se soucier des niveaux de gris des différentes structures. Cependant, on est souvent amené à travailler uniquement sur des structures claires ou des structures foncées d'une image. Ainsi, pour prendre en compte le niveau de gris dans le résultat des différents opérateurs précédemment présentés, nous introduisons un paramètre  $p$ , dont la valeur est comprise entre 0 et 1, qui pénalise les structures sombres au profit des structures claires :

$$p(x) = \frac{1}{M \times L} \sum_{x_i \in A_x^L} f(x_i), \quad (4.13)$$

avec  $M = \text{Card}(V)$  le nombre d'éléments dans l'ensemble des valeurs de  $V$ . Cette équation calcule donc la valeur moyenne des niveaux de gris du REGSE que l'on normalise par  $M$ , afin d'avoir une pénalisation comprise entre 0 et 1. Ainsi, on multiplie les valeurs des différents opérateurs par  $p$ , pour chaque pixel de l'image afin d'accorder plus d'importance aux structures claires.

Enfin, on rehausse les structures foncées de l'image en multipliant les valeurs des différents opérateurs par  $q$  :

$$q(x) = 1 - p(x). \quad (4.14)$$

On prend donc l'exemple de l'opérateur basé sur l'inertie qui est maintenant écrit de deux façons différentes :

$$\text{Ine}_{REGSE}^L(x) = \begin{cases} \frac{p(x)}{L} \sum_{x_i \in A_x^L} \|x_i - \bar{x}\|_{\mathcal{L}_2}^2, & \text{pour détecter les structures claires} \\ \frac{q(x)}{L} \sum_{x_i \in A_x^L} \|x_i - \bar{x}\|_{\mathcal{L}_2}^2, & \text{pour détecter les structures foncées} \end{cases} \quad (4.15)$$

Les résultats en images de ces opérateurs sont donnés dans la section suivante.

## 4.4 Résultats

Les REGSEs sont des éléments structurants mais ils ne sont pas seulement limités au domaine de la morphologie mathématique. Ils peuvent servir de noyaux de convolution pour créer des filtres linéaires. On prend l'exemple du filtre *moyenne*, qui attribue au pixel initial la valeur moyenne de l'élément structurant. On présente dans cette section des exemples de filtrages médians, moyens et alternés séquentiels pour la réduction du bruit, puis nous compareront les REGSEs avec les GANSEs et les amibes. Enfin, nous présenterons quelques résultats pour la détection d'éléments fins, pour le contrôle non destructif ou le rehaussement de fibres.

### 4.4.1 Exemple de filtres utilisant les REGSEs

On présente dans cette section, les résultats des filtres moyens, des filtres de rang et des filtres alternés séquentiels.

#### Filtres moyens

Par construction, les REGSEs ne franchiront pas les zones de fort gradient de l'image (à part si la contrainte de taille les y oblige) et seront constitués de pixels ayant une grande homogénéité par rapport au pixel initial. En calculant un filtre moyen avec des REGSEs, le rapport signal sur bruit va augmenter sans déplacer les zones de fort gradient de l'image. La figure 4.6(d) présente le résultat d'un filtre moyen avec des REGSEs (image pilote : filtre gaussien de taille 1) avec un zoom sur des détails de l'image cameraman. On remarque que le bruit, aussi bien dans le fond de l'image que dans l'avant plan, est considérablement réduit. De plus, les frontières ne sont pas déplacées et les petits détails inférieurs à 35 pixels (la taille des REGSEs) sont supprimés.

#### Filtres de rang et filtre médian

On peut aussi définir un filtre de rang, qui trie les valeurs du REGSE par ordre croissant et qui sélectionne une des valeurs de cette liste pour l'attribuer au pixel initial. Prendre la valeur minimale revient à calculer une érosion, prendre la valeur maximale est une dilatation et choisir une valeur intermédiaire comme la médiane, permet de construire un filtre médian. Le résultat de ce filtre est présenté dans la figure 4.6(e) et on remarque que la réduction du bruit est très bonne tout en conservant parfaitement les frontières, qui sont plus nettes que pour un filtre utilisant la moyenne.

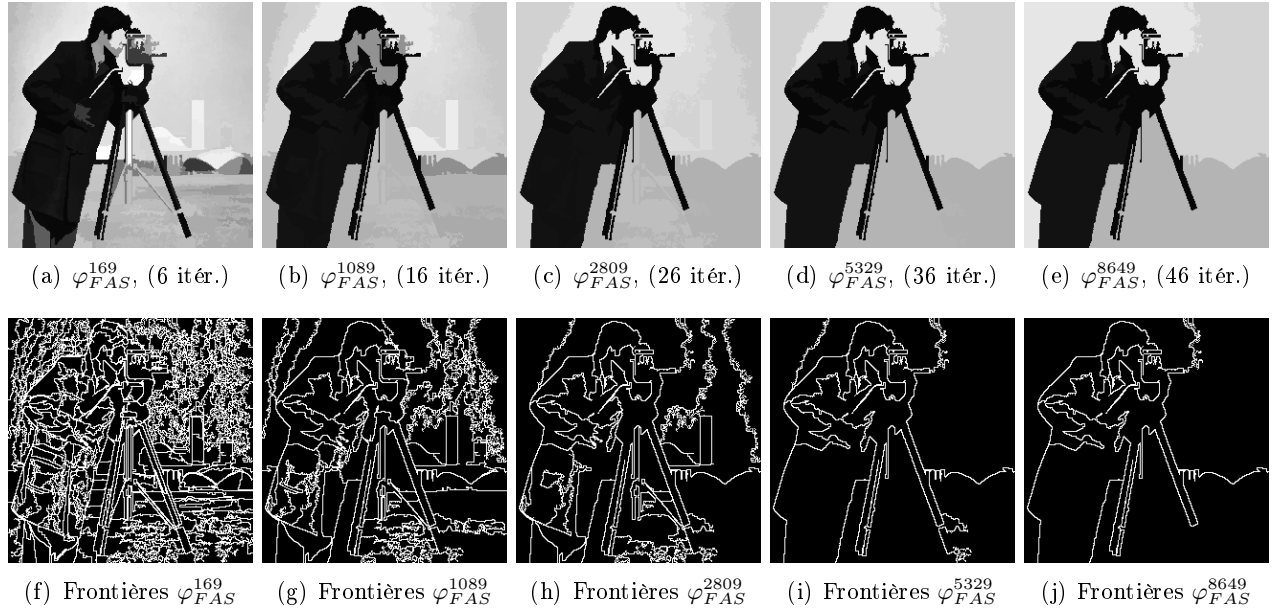


FIGURE 4.7 – Création de zones plates dans l'image avec des filtres alternés séquentiels  $\varphi_{FAS}^L$  de différentes tailles sans images pilotes. Le pas d'incrément de la taille de l'ES est de  $(2i + 1)^2$  (avec  $i$  l'itération numéro  $i$ ) entre chaque couple fermeture/ouverture. La première ligne (resp. la seconde) correspond aux résultats du filtrage (resp. aux frontières des zones plates).

### Filtres alternés séquentiels (FAS)

À partir des ouvertures et des fermetures par REGSE que nous avons développées dans la section 4.2.2, nous pouvons construire des filtres alternés séquentiels. C'est une notion introduite par Sternberg [219], puis reprise et intensivement étudiée par Serra dans [203]. Deux familles de filtres de tailles croissantes sont considérées et alternativement associées. On dénombre donc deux compositions de filtres à partir des ouvertures et des fermetures avec des REGSEs,  $\gamma_{FAS}^L$  et  $\varphi_{FAS}^L$  :

$$\gamma_{FAS}^L = \gamma^L \varphi^L \dots \gamma^2 \varphi^2 \gamma^1 \varphi^1, \quad (4.16)$$

$$\varphi_{FAS}^L = \varphi^L \gamma^L \dots \varphi^2 \gamma^2 \varphi^1 \gamma^1. \quad (4.17)$$

Les FAS permettent de simplifier l'image en créant des zones plates, en arasant successivement les pics et les vallées de tailles croissantes. La figure 4.6(f) le montre et permet de simplifier l'image tout en préservant correctement les frontières des régions plus grandes que le paramètre de taille. Les FAS permettent de réaliser une analyse multi-échelles et peuvent être utilisés pour segmenter l'image. La figure 4.7 présente une succession de FAS de tailles croissantes avec les frontières des zones plates ainsi générées.

#### 4.4.2 Comparaison avec d'autres méthodes

Dans cette section, nous comparons les opérateurs morphologiques basés sur les REGSEs avec les amibes morphologiques et les GANSEs respectivement introduits par Lerallut *et al.* et par Debayle et

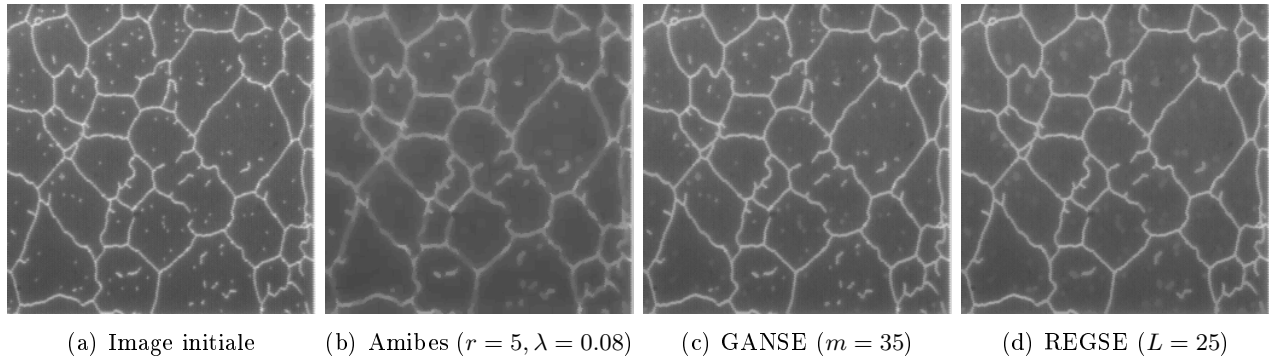


FIGURE 4.8 – Comparaison des ouvertures morphologiques avec différents éléments structurants.

Pinoli. Des explications sur le fonctionnement de ces deux méthodes sont données dans la section 4.1 tout comme la signification de leurs paramètres. Pour illustrer et comparer ces trois opérateurs, on utilise une image test (figure 4.8(a)) dont on souhaite filtrer les artefacts clairs au centre des cellules.

Le premier paramètre des amibes morphologiques est son rayon qui est fixé à 5 pixels. Le second paramètre  $\lambda$  permet de pondérer l'influence du gradient sur la forme. Ce n'est pas un paramètre facile à choisir puisqu'une trop faible pénalisation des gradients transforme ce filtre en un simple opérateur invariant par translation (le contenu de l'image n'est alors plus pris en compte). Au contraire, une trop forte valeur pour  $\lambda$  ne permettra pas à l'élément structurant de grandir laissant ainsi l'image initiale inchangée. Empiriquement, nous choisissons  $\lambda = 0,08$ . La figure 4.8(b) fournit le résultat pour des ouvertures avec des amibes morphologiques et on remarque que le bruit est considérablement supprimé. Néanmoins, les frontières sont floutées et les artefacts ne sont pas supprimés.

La méthode utilisant les GANSEs nécessite l'utilisation d'un unique paramètre  $m$ . Si  $m$  est égal à 0, alors le GANSE est incapable de franchir le moindre gradient et l'ES sera uniquement constitué de pixels connexes de même niveau de gris que le pixel initial. Ainsi, une trop faible valeur de  $m$  empêche la croissance de l'ES. Néanmoins, on note que la force du gradient est la même entre les artefacts et le fond de l'image qu'entre les frontières des cellules et le fond de l'image. Cela ne nous permet pas d'ajuster correctement la valeur de  $m$  et par conséquent une ouverture par GANSE est relativement inefficace pour supprimer ce genre d'artefacts (figure 4.8(c)).

Pour une ouverture utilisant les REGSEs, la paramétrisation est facile puisque l'unique paramètre contrôle le nombre de pixels de l'ES. Il faut donc fixer  $L$  en analysant la taille de la plus grande structure à supprimer. Dans notre exemple, on fixe  $L$  à 25 pixels pour correspondre à la taille maximale d'une amibe pour pouvoir plus facilement comparer ces deux opérateurs. On remarque sur la figure 4.8(d) que les frontières sont préservées tout en supprimant les artefacts.

**REMARQUE :** Une ouverture aréolaire aurait produit des résultats similaires.

D'un point de vue réduction du bruit, les amibes morphologiques ont un effet plus important que notre opérateur néanmoins, la contrainte de taille des REGSEs est un élément clé pour rehausser certaines caractéristiques de l'image, là où notre opérateur surpasse les GANSEs et les amibes.

#### 4.4.3 Application à la détection de fissures

Le projet TOCATA (cf. chapitre 7) a pour but la détection des défauts de surface de pièces métalliques. Les fissures sont des éléments difficiles à détecter du fait de leur faible largeur et l'idée première

de ce filtre vient de l'image de la figure 4.9(a). L'extraction de cette très fine fissure est possible si l'élément structurant adapte sa forme au contenu de l'image. Relâcher la contrainte de taille ajoute de la flexibilité aux ESs, qui peuvent suivre toutes les fissures sans restriction sur leur tortuosité. Ainsi, utiliser une ouverture par REGSE d'une taille correspondant approximativement à la longueur de la fissure à détecter permet de la rehausser. La figure 4.9 présente les résultats de la détection utilisant les opérateurs précédemment décrits : les amibes morphologiques, le supremum des ouvertures avec des segments, une ouverture avec des REGSEs et deux opérateurs analysant la forme des REGSEs.

Le supremum des ouvertures avec des segments orientés dans toutes les directions de l'espace, permet de filtrer toutes les structures qui ne sont pas rectilignes dans l'image. Ce filtre supprime le bruit mais il supprime aussi des éléments de la fissure lorsque ceux ci deviennent trop tortueux. Quant aux amibes morphologiques, elles ne sont pas adaptées à ce type de structures filiformes et proposent une détection beaucoup moins bonne que pour les ouvertures avec des REGSEs. En effet, pour une taille  $L = 150$  pixels, le bruit est supprimé tout en préservant parfaitement la fissure.

Plus intéressant, les nouveaux opérateurs basés sur l'analyse de la forme des REGSEs proposent une détection efficace des éléments fins correspondant à la fissure. Les opérateurs utilisant le moment d'inertie et l'opérateur d'extension maximale (en pénalisant les structures sombres, voir section 4.3) proposent des résultats directement exploitables puisqu'un simple seuillage permet d'isoler les fissures. On note que la structure circulaire qui était préservée par les ouvertures avec des REGSEs est maintenant supprimée par ces opérateurs.

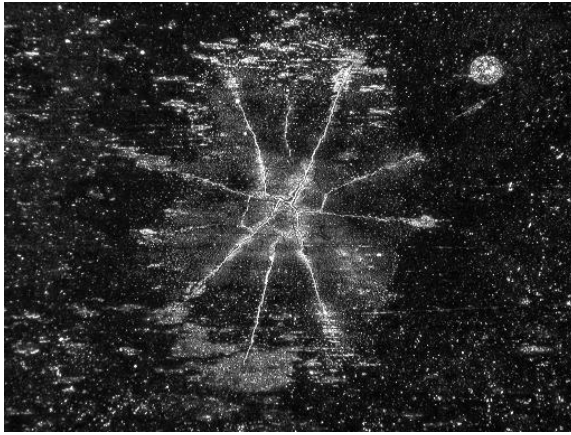
### 4.4.4 Application au rehaussement de fibres

Le projet L'Oréal, chapitre 8, propose une caractérisation des fibres du derme de la peau humaine, qui sont donc des éléments filiformes et de profil gaussien. Ainsi, lorsque le point initial se situe sur une de ces fibres, l'élément structurant va se déformer afin d'adapter sa forme. Premièrement, on présente les résultats des opérateurs qui analysent la forme des REGSEs, sur des fibres synthétiques puis sur des fibres de collagène.

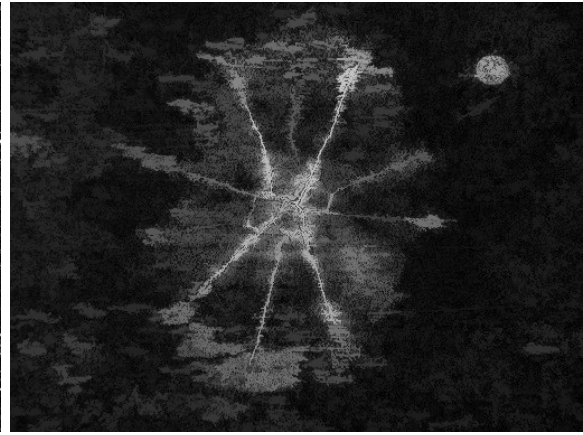
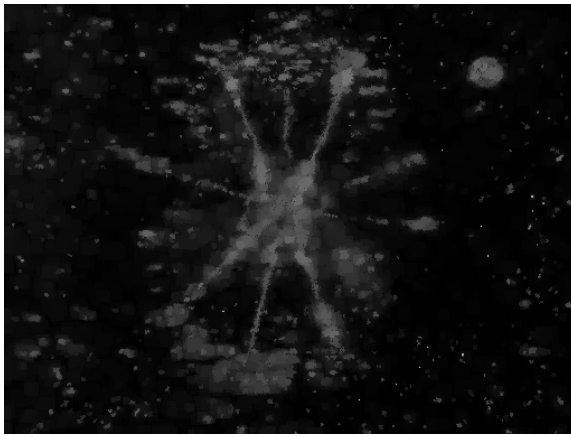
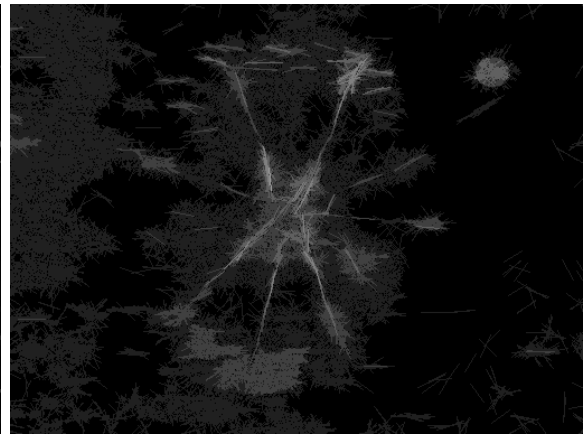
#### Fibres d'un modèle aléatoire

On génère un modèle aléatoire, avec comme grain primaire une fibre tortueuse de taille comprise entre 3 et 80 pixels, en suivant le protocole décrit dans la section 3.6.2. Nous obtenons une image de fibres qui ont toutes un profil gaussien et une longueur variable. Nous appliquons à cette image les trois opérateurs développés dans la section 4.3 : les opérateurs d'inertie, d'extension maximale et d'excentricité. Les résultats sont regroupés dans la figure 4.10 et voici quelques remarques :

- Pour les résultats des opérateurs basés sur l'inertie et sur l'extension maximale, on remarque que le niveau de gris est fortement corrélé à la taille de la structure. En effet, on peut trier les fibres selon leur taille en seuillant l'image (sous réserve que les fibres aient un niveau de gris identique dans l'image initiale).
- La différence entre  $\text{Ine}_{REGSE}^L$  et  $\text{Ext}_{REGSE}^L$  est relativement grande puisque l'inertie du REGSE augmente plus rapidement avec la taille de l'ES (la distance est élevée au carré dans l'équation 4.10). On a donc tendance à rehausser en priorité les très grandes fibres de l'image avec l'opérateur inertiel.
- Lorsque l'origine est sur une extrémité d'une fibre, l'ES va se propager à l'intérieur et son bary-centre sera éloigné du point initial. Par conséquent, l'opérateur d'excentricité permet de rehausser



(a) Image Initiale

(b) Ouverture avec des REGSEs ( $L = 150$  pixels)(c) Amibes morphologiques (rayon de 5 pixels,  $\lambda = 0.025$ )

(d) Ouverture avec des éléments structurants linéaires de longueur 21 pixels. Les lignes sont orientées tous les degrés.

(e) Opérateur inertiel avec des REGSEs, pénalisant les structures sombres ( $L = 80$  pixels)(f) Opérateur d'extension maximale avec des REGSEs, pénalisant les structures sombres, ( $L = 80$  pixels)

FIGURE 4.9 – Résultats de l'extraction des défauts de surface avec différentes méthodes.



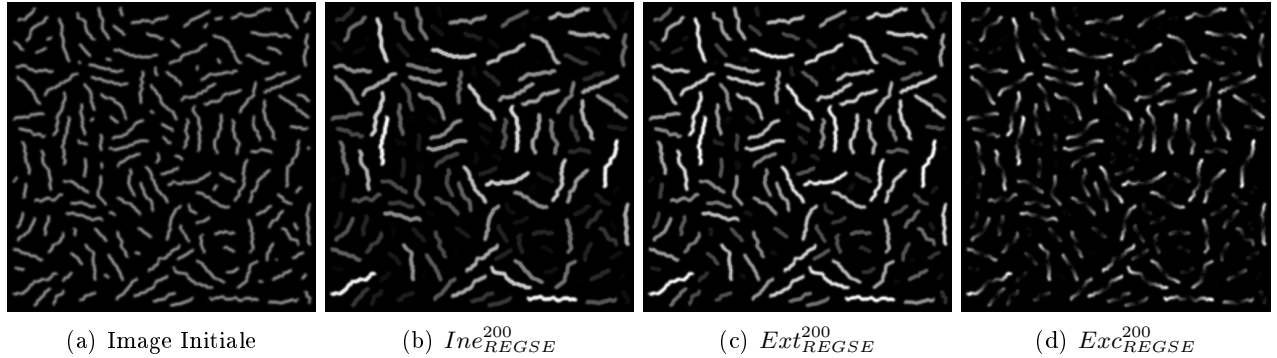


FIGURE 4.10 – Résultats des opérateurs basés sur la forme des REGSEs sur une réalisation d'un modèle aléatoire, avec comme grain primaire une fibre tortueuse.

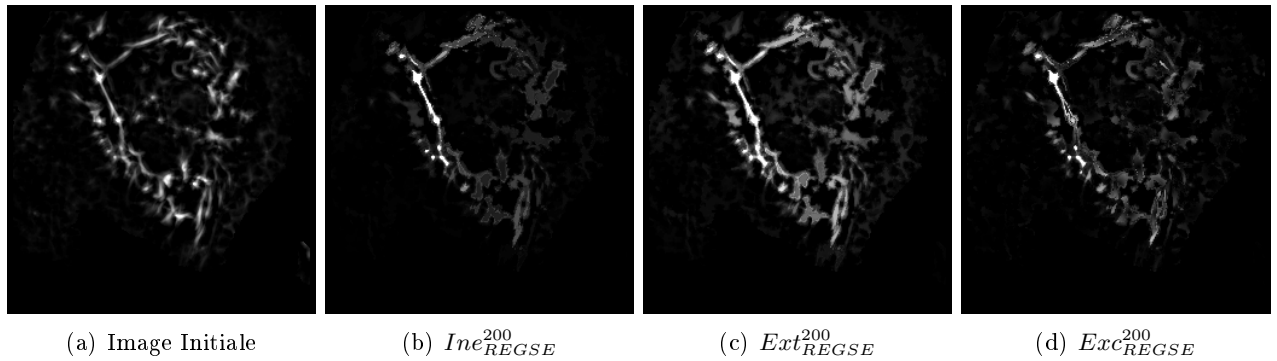


FIGURE 4.11 – Résultats des opérateurs basés sur la forme des REGSEs sur une image d'une coupe d'un volume 3-D, d'un échantillon de la peau.

les points terminaux des fibres et, si on applique par la suite un opérateur de type *h-maxima*, on pourra localiser l'ensemble des points terminaux.

### Fibres de collagène du derme

Les fibres du derme de la peau ont été préalablement rehaussées par un filtre *Difference of Gaussians*. Le chapitre 8 donne plus d'explications sur ce filtre et on remarque le profil gaussien des fibres. On applique donc les trois mêmes opérateurs sur une coupe du volume 3-D et la figure 4.11 regroupe les résultats obtenus. Avec ces opérateurs, on rehausse différentes fibres en fonction de leur longueur et de leur forme. De plus, avec un opérateur d'extension maximale ou d'inertie, on obtient des informations sur la distribution de taille des fibres présentes dans l'image.

## 4.5 Considérations pratiques

La construction d'un REGSE utilise un procédé de croissance de régions cependant, lorsqu'il faut calculer un REGSE pour chaque pixel de l'image, il est nécessaire d'écrire une implémentation rapide.

L'équation 4.3 décrit une construction récursive de l'élément structurant mais il ne s'agit pas simplement d'utiliser un algorithme récursif d'un point de vue informatique. En effet, il faut utiliser une structure de données adaptée pour simuler la récursivité. Dans cette section, on décrit l'implémentation utilisée, sa complexité et ses temps d'exécution.

### 4.5.1 Implémentation de l'algorithme

#### Structure de données

La structure de données utilisée est une file d'attente hiérarchique (FAH). Cette FAH est composée de  $M$  files (une pour chaque niveau de gris) classées par priorité décroissante. Ainsi,  $FAH(0)$  (resp.  $FAH(M)$ ) signifie la file d'attente de plus haute priorité (resp. de plus basse priorité). L'insertion d'un pixel  $x_i$  dans la FAH va se faire avec une priorité  $\rho(x, x_i)$  qui dépend de  $x$ , le pixel initial. Par conséquent, l'opération *insérer un pixel  $x_i$  dans la file d'attente hiérarchique* est écrite  $FAH(\rho(x, x_i)).push(x_i)$  et l'opération inverse, *retirer de la file d'attente hiérarchique l'élément de plus haute priorité* est réalisée avec un *pop*, décrit dans l'algorithme 8. Cette structure de données s'adapte bien à notre problématique et nous évite les opérations de tris des données qui se révèlent être très coûteuses en temps de calcul.

---

**Algorithme 8:** Extraction d'un élément de la FAH :  $y = Pop(FAH)$

---

**Input :**  $FAH$  : la file d'attente hiérarchique

**Result :**  $y$ , l'élément de priorité maximale

```

1 Prérequis : la FAH n'est pas vide
2 for  $i = 0 \rightarrow M$  do
3   if  $FAH(i)$  n'est pas vide then
4     return  $FAH(i).pop()$ 
```

---

#### Implémentation

Le calcul d'un REGSE est réalisé avec un procédé classique de croissance de région. L'algorithme 9 montre comment on extrait les  $L$  pixels formant  $A^L$ . On commence par insérer le pixel initial dans la pile à la priorité maximale (ligne 4) et on marque ce pixel comme traité (ligne 5). Puis, dans la boucle (lignes 6 à 15), on dépile le pixel ayant la plus grande priorité puis on insère ses pixels voisins avec une priorité  $\rho(p, x)$ , s'ils n'ont pas déjà été empilés. Ainsi, on considère les  $L$  premiers pixels qui sont dépilés par la file d'attente hiérarchique (ligne 8) comme élément structurant. Lorsque le nombre d'éléments à insérer dans  $A^L$  est atteint, le processus se termine et  $A^L$  est renvoyé à la fonction appelante (ligne 10).

#### Optimisation de l'algorithme

L'optimisation de cet algorithme est possible et permet d'améliorer les performances en termes de temps de calcul. On note premièrement que l'ES n'est pas séparable comme pourrait l'être le noyau d'un filtre gaussien ou les ESs rectangulaires d'une érosion. Aussi, on détaille ici les astuces de programmation utilisées :

#### 4. REGSE

---

**Algorithme 9:** Calcul d'un REGSE :  $A^L = \text{computeREGSE}(f, x, L)$

---

**Input** :  $f$  : l'image pilote  
 $x$  : pixel initial  
 $L$  : taille de l'élément structurant  
**Result** :  $A^L$  : l'élément structurant REGSE

- 1 **Prérequis** :  $L \geq 1, x \in D, \text{Card}(D) \geq L$
- 2  $State \leftarrow NOT\_PROCESS$
- 3  $A^L \leftarrow \emptyset$
- 4  $FAH(0).push(x)$
- 5  $State(x) \leftarrow PROCESS$
- 6 **repeat**
- 7      $y \leftarrow Pop(FAH)$
- 8      $A^L \leftarrow A^L \cup y$
- 9     **if**  $\text{card}(A^L) == L$  **then**
- 10         **return**  $A^L$
- 11     **foreach**  $p \in \mathcal{N}(y)$  **do**
- 12         **if**  $State(p) == NOT\_PROCESS$  **then**
- 13              $State(p) \leftarrow PROCESS$
- 14              $FAH(\rho(p, x)).push(p)$
- 15 **until** 1     \\ Boucle infinie

---

- **Optimisation de la FAH** : à partir de la taille de l'élément structurant, on peut déduire la taille maximale nécessaire pour la file d'attente hiérarchique. En effet, il ne peut y avoir au maximum que  $4L$  pixels dans chaque file d'attente (À chaque itération, un pixel est dépilé et au maximum, 5 pixels voisins peuvent être insérés dans la même file). L'allocation de la mémoire de la FAH est donc réalisée une unique fois au début de l'opérateur supprimant les allocations et les désallocations superflues. La FAH est donc composée de  $4LM$  places qui sont stockées dans un tableau 1-D. Pour obtenir le même comportement qu'une pile classique, on stocke, pour chaque pile, la position du premier pixel  $sp$  et la position du dernier pixel  $ep$  dans ce buffer circulaire. La taille de cette file est donnée par la soustraction de  $ep$  et de  $sp$ , et supprimer tous les éléments de la pile revient simplement à réinitialiser  $sp$  et  $ep$  à zéro.  $sp$  et  $ep$  sont donc des tableaux de  $M$  valeurs.
- **Optimisation d'un Pop** : pour connaître rapidement la file d'attente de priorité maximale qui n'est pas vide, on stocke le numéro de cette file et on le met à jour à chaque fois l'on insère un nouveau pixel ou que l'on fait un  $Pop()$ . Cela permet d'éviter des balayages superflus de toutes les piles comme le fait l'algorithme 8.
- **Optimisation de State** : on étend la variable  $State$ , qui est maintenant une représentation de  $D$  dans  $\mathbb{N}$ . Pour ne pas réinitialiser cette variable à zéro avant le calcul de chaque REGSE (ligne 1 de l'algorithme 9), chaque REGSE sera représenté par une clef unique, sa position dans l'image. Ainsi, pour le pixel initial  $x_i$ , les lignes  $State(x_i) \leftarrow PROCESS$  et  $State(p) == NOT\_PROCESS$  seront remplacées par  $State(x_i) \leftarrow i$  et par  $State(p) \neq i$ .

### Approximation de l'algorithme

L'accélération de cet algorithme passe aussi par la création d'une approximation de ces opérateurs. On considère alors que deux pixels voisins dans l'image, ayant le même niveau de gris, auront le même résultat. Il n'est donc pas nécessaire d'extraire un REGSE pour chaque pixel de l'image, accélérant ainsi cet algorithme. Pour des images binaires, les gains en temps de calcul sont très grands et cette approximation fournit des résultats très similaires. Les résultats seront même identiques pour des ouvertures ou des fermetures. En effet, on a vu que les REGSEs agrègent prioritairement les pixels des régions plates, puis débordent si la taille de cette composante connexe est plus petite que  $L$ . Ainsi tous les pixels de cette CC auront le même minimum et le même maximum. Pour des images en niveaux de gris, l'accélération obtenue dépend du contenu de l'image mais on note que cette approximation fournit des résultats où les différences sont imperceptibles à l'œil.

Avec tous les éléments présentés dans cette section, les temps de calcul sont considérablement réduits. La parallélisation de ces opérateurs est simple et permet de profiter pleinement de la puissance des ordinateurs récents. Les temps de calcul seront présentés dans la section 4.5.3. Néanmoins, seules des images 8 bits peuvent être traitées efficacement avec l'algorithme présenté. En effet, il utilise une file d'attente hiérarchique qui est inefficace avec des précisions supérieures à 8 bits. Il faudrait donc utiliser des files d'attente à priorité : *priority queue*.

#### 4.5.2 Complexité algorithmique

La complexité algorithmique des opérateurs basés sur des REGSEs peut se décomposer en deux analyses : la complexité de l'extraction du REGSE et la complexité de l'opération (recherche de la valeur minimum, maximum, moyenne, médiane, calcul de l'inertie...).

La construction d'un élément structurant par un processus de croissance de régions (algorithme 9) nécessite au total  $L$  itérations, correspondant à l'agrégation des  $L$  pixels de l'ES adaptatif. Lors de chaque itération, on insère au maximum 5 pixels et chaque insertion d'un élément dans la FAH est réalisée en temps constant. La fonction *Pop* est utilisée exactement  $L$  fois et à une complexité égale à  $\mathcal{O}(L \times \log(M))$  avec  $M = \text{card}(V)$ . Par conséquent, la complexité algorithmique pour la construction d'un REGSE est de  $\mathcal{O}(L \times \log(M))$ .

Une fois la forme de l'élément structurant obtenue, il reste à calculer les différentes opérations spécifiques à un opérateur donné. Pour une érosion, on cherche la valeur minimale de l'ES et cette opération est réalisée simultanément à la construction du REGSE. Ainsi, la complexité est de  $\mathcal{O}(L)$  pour une érosion, pour une dilatation ou pour un filtre moyen. Pour les filtres de rang, on utilise un histogramme pour accéder plus facilement au classement des différentes valeurs et la complexité passe en  $\mathcal{O}(L \times M)$ . Enfin, pour les opérateurs qui analysent la forme de l'élément structurant, le calcul du barycentre est indispensable avant de calculer la valeur d'un attribut. Ainsi, il faut parcourir une seconde fois les pixels de l'ES, que auront été préalablement stockés dans un tableau. Cependant, ce parcours supplémentaire ne modifie pas la complexité de cette opération,  $\mathcal{O}(L)$ .

Ainsi, la complexité globale pour une érosion avec des REGSEs est de :

$$\mathcal{O}(\text{Width} \times \text{Height} \times L \times \log(M)), \quad (4.18)$$

avec *Width* et *Height*, la largeur et la hauteur de l'image.

## 4. REGSE

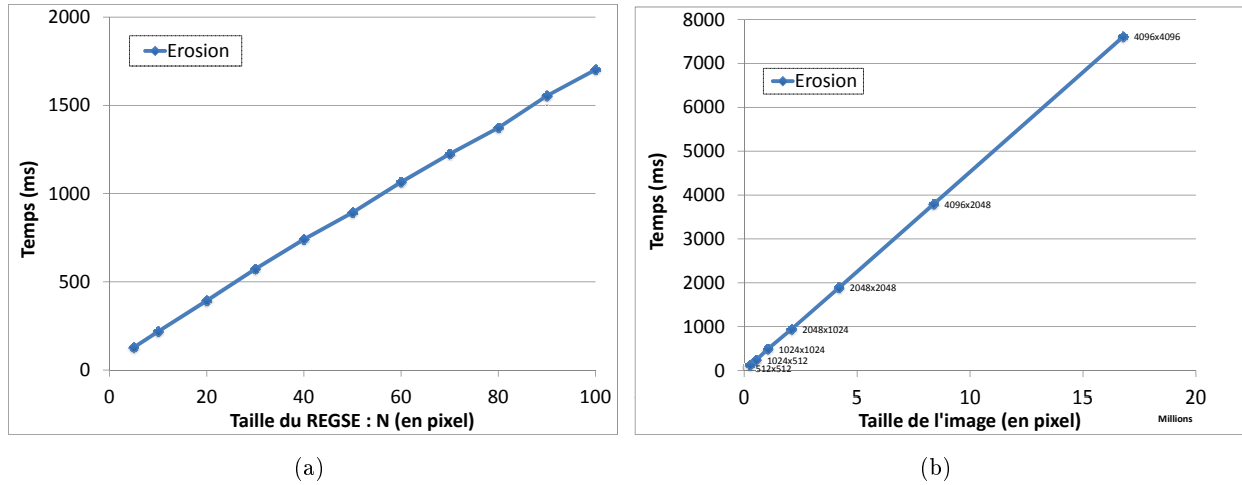


FIGURE 4.12 – Temps de calcul pour une érosion avec des REGSEs en fonction de, (a) la taille de l'élément structurant (image goldhill 512x512 pixels), (b) la taille de l'image ( $L = 5$  pixels, image goldhill répliquée pour ne pas modifier les structures de l'image).

### 4.5.3 Temps de calcul

On développe dans cette section des benchmarks testant différents paramètres de ces opérateurs basés sur des REGSEs. Le premier test propose de montrer que les temps de calcul sont linéaires avec la taille de l'image ainsi qu'avec la taille de l'ES. Puis nous verrons l'intérêt de paralléliser l'algorithme ou d'utiliser l'approximation définie dans la section 4.5.1. Enfin, nous comparerons les temps d'exécution entre les différents opérateurs basés sur les REGSEs puis nous les comparerons avec les amibes morphologiques et les GANSEs. Tous ces tests sont réalisés sur un ordinateur portable intel core (TM) i7-2860QM CPU @ 2.50GHz.

#### Temps de calcul en fonction de $L$ et de la taille de l'image

La section 4.5.2 illustre que la complexité d'une érosion avec des REGSEs est, à la fois linéaire en fonction de  $L$  et en fonction de la taille de l'image (à une constante  $\log(M)$  près). On calcule donc les temps moyens pour 10 érosions avec des REGSEs sur l'image goldhill et sur des versions répliquées de goldhill (pour former des images de différentes tailles). Les figures 4.12(a) et 4.12(b) présentent les résultats des deux expérimentations et on constate une correspondance entre les temps de calcul et la complexité théorique.

#### Parallélisation de l'algorithme et influence de l'approximation

De la même manière que l'on peut paralléliser une érosion classique, on peut diviser l'image en bandes de tailles identiques et donner à chaque thread du processeur un pointeur sur l'image entière et un ensemble de pixels à traiter. La figure 4.13(a) donne le facteur d'accélération obtenu en fonction du nombre de threads utilisés pour l'érosion avec des REGSEs de l'image goldhill, 512 par 512 pixels. Les temps de calcul sont donc divisés par 1,96, si on utilise deux threads ou par 4,26 avec 8 threads.

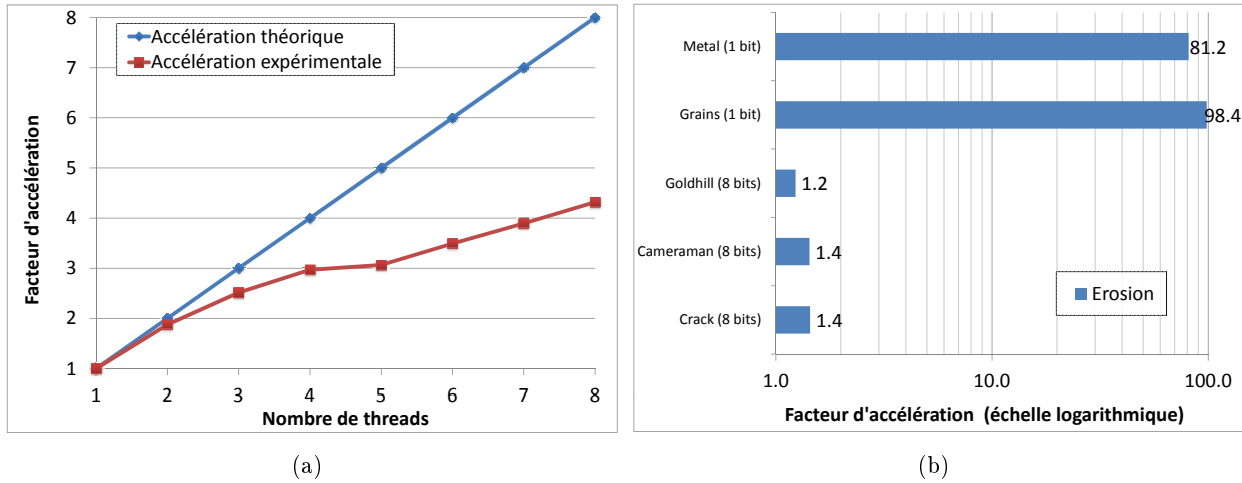


FIGURE 4.13 – Gain apporté par : (a) la parallélisation de l'algorithme (image Goldhill) et (b) l'approximation définie dans la section 4.5.1.

Pour mesurer l'apport de l'approximation définie dans la section 4.5.1, on calcule le facteur d'accélération entre les temps d'exécution avec et sans approximation, pour des images ayant un contenu différent. La figure 4.13(b) regroupe les résultats et on remarque que les temps de calcul ont été divisés par  $\approx 90$  pour des images binaires (*metal* et *grains*). On note de plus, que cette approximation permet de réduire les temps de calcul de 20 à 30% pour des images en niveaux de gris.

### Temps de calcul et comparaison des méthodes

Pour compléter nos tests sur les temps d'exécution, nous comparons dans un premier temps, les différents opérateurs basés sur les REGSEs que nous avons développés dans ce chapitre. La figure 4.14(a) présente les temps d'exécution moyens pour différents opérateurs : une érosion, une dilatation, les filtres moyen et médian ainsi que les trois opérateurs analysant la forme de l'élément structurant, l'inertie, l'extension maximale et l'excentricité. Les temps de calcul des différents opérateurs sont conformes à nos attentes puisque les opérateurs d'inertie et d'extension maximale sont plus lents. En effet, une fois que l'on a obtenu la forme de l'ES, il est nécessaire de le balayer une fois de plus, pour calculer son inertie ou son extension maximale. Pour l'opérateur d'excentricité, ce parcours supplémentaire n'est pas réalisé puisque l'on compare uniquement le barycentre au point initial.

Enfin, nous analysons les différences des temps d'exécution pour le calcul d'une érosion avec des REGSEs, des amibes morphologiques ou des GANSEs. Néanmoins, cette comparaison n'est pas aisée puisque les paramètres des algorithmes n'ont pas la même signification ni la même influence sur les temps de calcul. Nous fixons donc la taille maximale de l'amibe à la taille de l'élément structurant REGSE,  $L = (2r + 1)^2$ , avec  $r$  le rayon de l'amibe, puis nous choisissons des valeurs de  $m$  allant de 1 à 8 niveaux de gris pour les GANSEs. Enfin, les amibes morphologiques ont besoin d'un second paramètre, la pénalité des gradients ( $\lambda$ ), qui a une influence forte sur les temps de calcul. Pour  $\lambda = 0$ , aucune pénalité n'est associée aux gradients et les ESs ont une forme fixe (le calcul est lent puisque la propagation doit toujours atteindre la taille maximale de l'amibe). Pour  $\lambda > 0$ , la taille de l'élément

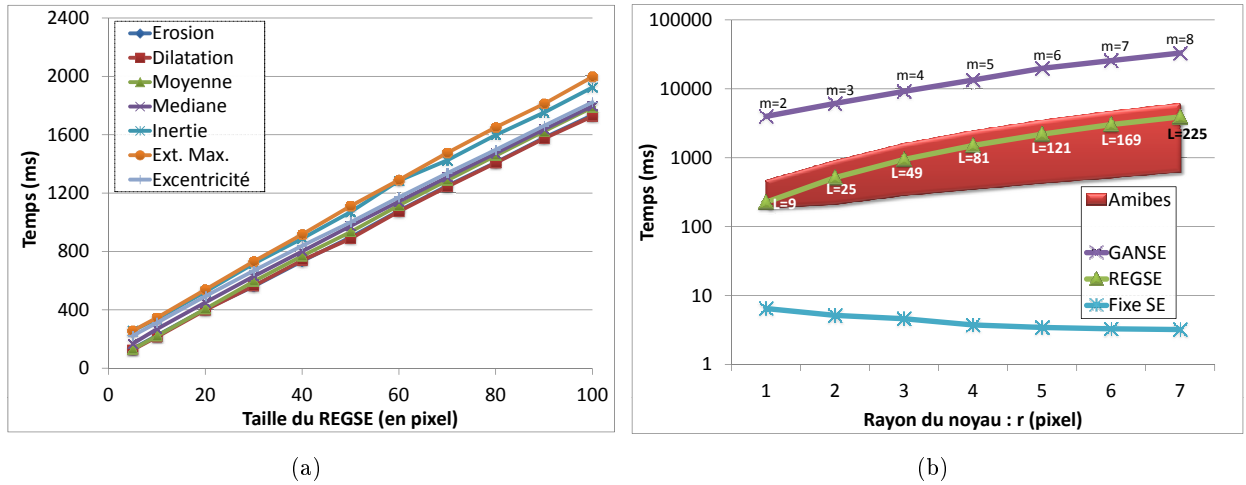


FIGURE 4.14 – Comparaison des temps de calcul entre (a) les différents opérateurs basés sur les REGSEs, (b) les érosions de différentes méthodes. Image Goldhill 512x512 pixels

structurant diminue jusqu'à ce que l'ES soit réduit à son pixel central (le calcul est alors très rapide puisque la propagation est stoppée dès que le pixel initial est inséré dans l'élément structurant).

La figure 4.14(b) résume tous les résultats des 3 opérateurs précédemment décrits. Pour les amibes, on obtient un large intervalle de valeurs où les valeurs maximales des temps (respectivement valeurs minimales) correspondent à  $\lambda = 0$  (respectivement  $\lambda = 2$ ). Comme point de repère, nous avons ajouté les temps de calcul d'une érosion avec un élément structurant fixe (carré). Ce dernier algorithme est très optimisé puisqu'il est décomposé en deux ES linéaires qui sont eux mêmes calculés en temps constant par pixel (voir chapitre 2). Pour des éléments structurants ayant une forme fixe mais arbitraire, une implémentation rapide consiste à utiliser des histogrammes [240]. On remarque que les temps de calcul d'une érosion avec des éléments structurants adaptatifs sont plus lents de deux ordres de grandeurs. De plus, les amibes et les REGSEs ont la même complexité et sensiblement les mêmes temps de calcul. Les GANSEs sont des opérateurs très lents puisque les ES n'ont pas de contrainte de taille et peuvent recouvrir toute l'image.

## 4.6 Bilan et perspectives

Ce chapitre présente un nouvel élément structurant qui s'adapte au contenu de l'image pour construire des opérateurs linéaires ou non linéaires. Ces REGSEs ont une taille fixe mais une forme libre. L'unique contrainte sur la forme de l'ES porte sur sa connexité : les REGSEs doivent former un élément structurant connexe. Avec les REGSEs, on peut utiliser un nombre important d'opérateurs de morphologie mathématique, dès lors qu'ils utilisent un élément structurant. Les ouvertures et fermetures avec des REGSEs offrent des possibilités de filtrage fin, puisqu'ils préservent les frontières de l'image si la structure analysée est plus grande que le paramètre de ce filtre. Cet unique paramètre contrôle le nombre de pixels de l'ES, offrant une paramétrisation simple et intuitive.

Les REGSEs utilisent un processus de croissance de régions pour leur construction et par conséquent, l'extension de ces opérateurs à des images 3-D ou plus est directe et nécessite uniquement un changement de connectivité.

Cette souplesse dans la forme leur permet de suivre toutes les structures filiformes de l'image, quelle que soit leur tortuosité. C'est l'attrait principal de ce filtre puisque l'on pourra alors analyser la forme de l'élément structurant pour rehausser les structures fines de l'image. À notre connaissance, c'est la première fois que la forme des éléments structurants adaptatifs est analysée pour obtenir de l'information. Les opérateurs d'inertie et d'extension maximale ont permis de rehausser les fissures de l'image pour réaliser un contrôle non destructif de pièces métalliques.

On peut généraliser l'analyse de la forme de l'élément structurant à d'autres méthodes ; tous les éléments structurants adaptatifs peuvent donc utiliser ces opérateurs (GANSE, amibes,...). On propose d'utiliser les éléments structurants basés sur les zones quasi plates de l'image. On appelle une zone plate<sup>1</sup>, toute région connexe de l'image composée de pixels ayant le même niveau de gris et n'étant pas contenue dans une autre zone plate. Calculer toutes les zones plates permet d'obtenir une unique partition de l'image. Les  $\lambda$ -zones plates ont été introduites en 1979 par Nagao *et al.* [162] et sont aussi appelés zones quasi plates [144, 145, 143] ou zones  $\alpha$ -plates [208, 212]. Toutes ces notions font référence à la même définition : deux pixels d'une image en niveaux de gris sont connectés s'il existe un chemin entre ces deux pixels tel que, la valeur absolue de la différence des niveaux de gris entre les pixels adjacents du chemin n'excède pas un seuil donné :  $\lambda$ . Les  $\lambda$ -zones plates permettent d'obtenir une unique partition de l'image et sont largement utilisées pour des tâches de segmentation [141, 214, 249].

On utilise les  $\lambda$ -zones plates comme éléments structurants, appelés  $\lambda$ -FZSE.

La figure 4.15 montre sur un signal 1-D, la différence entre un élément structurant GANSE et un  $\lambda$ -FZSE. Les éléments structurants, basés sur les  $\lambda$ -zones plates sont *réflexifs* (le pixel initial appartient à l'ES), *symétriques* (si  $x$  est connecté à  $y$ , alors  $y$  est connecté à  $x$ ) et *transitifs* (si  $x$  est connecté à  $y$  et  $y$  est connecté à  $z$ , alors  $x$  est connecté à  $z$ ). Ces trois propriétés sont intéressantes puisque cela signifie que l'élément structurant sera le même pour tous les pixels appartenant à une  $\lambda$ -zones plates. Il n'est donc pas nécessaire de calculer la forme de l'élément structurant pour tous les pixels de l'image, ce qui accélère considérablement cet algorithme (la complexité d'un tel algorithme est linéaire par rapport au nombre de pixels de l'image). Pour l'image goldhill que l'on a utilisée pour nos benchmarks (section 4.5.3), 13 ms sont nécessaires pour le calcul d'une érosion avec des  $\lambda$ -FZSE et on note que les temps de calcul sont constants avec l'unique paramètre de ce filtre,  $\lambda$ .

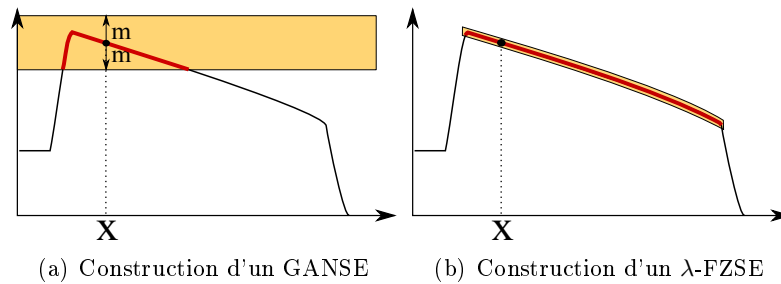


FIGURE 4.15 – Construction d'un élément structurant adaptatif à partir du point initial  $x$  sur un signal 1-D. (a) GANSE, (b)  $\lambda$ -FZSE.

Pour des images en niveaux de gris, les éléments structurants vont se propager dans les objets fins et ils vont pouvoir les suivre sans restriction sur leur tortuosité. L'analyse de la forme de l'élément

1. zone plate : *flat zone* en anglais



#### 4. REGSE

---

structurant ainsi obtenu permettra de faire la distinction entre une structure filiforme et les autres. La figure 4.16 présente les résultats obtenus sur une image possédant une fissure à détecter. On remarque que les résultats obtenus avec des  $\lambda$ -FZSEs sont très bon avec des temps de calcul bien plus rapides qu'avec des GANSEs. Cet opérateur sera donc utilisé dans le projet TOCATA (chapitre 7), puisqu'il allie vitesse et pouvoir de détection.

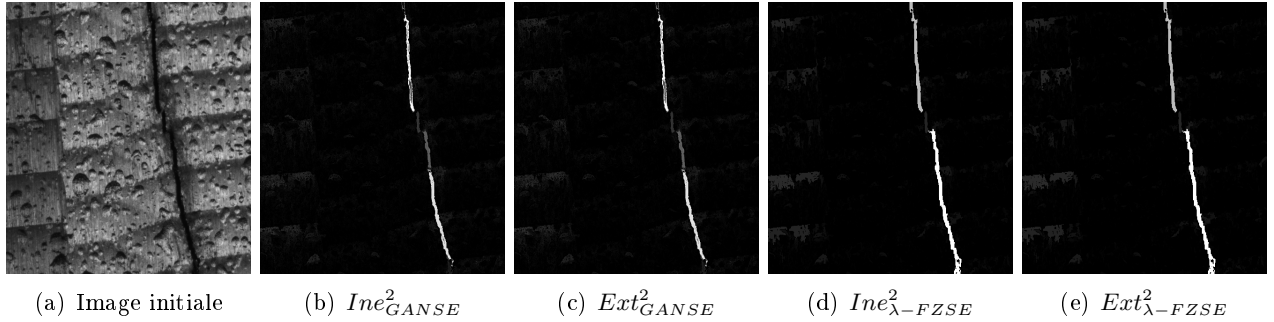


FIGURE 4.16 – Illustration des opérateurs d'inertie et d'extension maximale avec des GANSEs et des  $\lambda$ -FZSEs.

Un parallèle peut être mené entre l'analyse de la forme des éléments structurants et l'analyse des composantes connexes des opérateurs par attributs. Cependant, les objets analysés par ces deux méthodes sont très différents et fournissent des informations complémentaires. Aussi, dans le chapitre suivant, nous introduisons un filtre connexe utilisant de nouveaux attributs géodésiques et topologiques afin d'obtenir une bonne représentation de la longueur des structures.

Enfin, pour étendre l'utilisation des opérateurs avec des REGSEs, nous souhaitons le rendre plus robuste au bruit. En effet, le bruit présent dans l'image peut déconnecter certaines structures filiformes et altérer leur détection. Nous modifierons donc la fonction d'homogénéisation  $\rho$  en ajoutant un second terme dans cette équation qui augmentera fortement la priorité d'un pixel bruité, si ce dernier possède un voisin très homogène par rapport au pixel initial et pas encore traité.

---

## Filtres connexes : nouveaux filtres permettant de calculer des amincissements par attributs géodésiques et géométriques

---

Détecter des structures filiformes et les extraire d'une image peut être réalisés en analysant des longueurs de segments de droites (chapitre 2) ou des longueurs de chemins (chapitre 3). Ces opérateurs sont moins précis si la tortuosité des structures à détecter est trop importante. Les REGSEs sont des éléments structurants adaptatifs qui peuvent suivre toutes les structures fines de l'image. Ces opérateurs proposent une détection efficace qui peut cependant être améliorée si on travaille composante connexe par composante connexe. On propose donc, dans ce chapitre, d'associer un opérateur connexe qui préserve les contours de l'image, avec des nouveaux attributs pour former des *amincissements par attributs géodésiques et géométriques*.

Une ouverture par attribut est un opérateur anti-extensif, croissant et idempotent. Il supprime de l'image toutes les composantes connexes qui ne respectent pas un critère donné. Si un filtre n'est pas croissant, alors on utilise le terme plus général d'amincissement à la place d'ouverture. Ce chapitre propose donc d'utiliser des critères géodésiques et géométriques non croissants pour construire des amincissements pour des images en niveaux de gris.

Le diamètre géodésique permet d'avoir une bonne représentation de la longueur d'une structure mais son calcul est relativement lent. Ainsi, on développe un nouvel attribut géodésique : *le diamètre barycentrique*. Ces deux attributs ont un objectif commun, mesurer la longueur d'un objet. Cependant, le calcul du diamètre barycentrique est réalisé d'une manière efficace, en temps constant par pixel. On donnera alors l'erreur moyenne et l'erreur maximale théorique entre ces deux mesures et on remarquera que dans la pratique, les résultats sont très similaires. D'autres optimisations algorithmiques sont présentées de ce chapitre afin d'obtenir des accélérations de plusieurs ordres de grandeur par rapport à une implémentation naïve.

La taille des structures peut aussi être mesurée en calculant la longueur de l'axe médian. On obtient donc des amincissements par attributs géométriques.

On illustre l'utilisation de ces filtres sur des images du projet TOCATA (chapitre 7), pour la détection des défauts de surface de type fissure. On discutera des avantages et des inconvénients de ces filtres en les comparant avec les autres méthodes développées dans les chapitres précédents.

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSQUES ET GÉOMÉTRIQUES

Mots clés : Morphologie mathématique, attributs géodésiques, attributs géométriques, diamètre géodésique, diamètre barycentrique, axe médian, détection des structures fines, élongation, tortuosité, amincissement, épaississement.

Ce travail a en partie donné lieu à deux publications :

1. Dans Morard *et al.* [150], on introduit les amincissements par attributs géodésiques dans une conférence internationale :  
**V. Morard, E. Decencière, and P. Dokládal, Geodesic attributes thinnings and thickenings, *Proceedings of the 10th International Conference on Mathematical Morphology and its Applications to Image and Signal Processing (ISMM)*, Verbania, Italy, 2011**
2. Ce travail a été par la suite, étendu et publié dans le *Journal of mathematics, imaging and vision* (JMIV). Dans [152], l'accent a été mis sur les temps de calcul puisque l'on introduit le diamètre barycentrique et des optimisations diverses sur l'algorithme :  
**V. Morard, E. Decencière, and P. Dokládal, Efficient geodesic attributes thinnings based on the barycentric diameter, *Journal of Mathematics Imaging and Vision, JMIV*, 2012**

### 5.1 État de l'art

Les techniques de contrôle non destructif ont pour objectif d'inspecter des pièces, des tissus, des œuvres d'art ou d'autres objets à la recherche d'éventuels défauts et ce, sans les détériorer. La recherche de fissures sur des pièces métalliques industrielles avec un système de vision (chapitre 7) nécessite des opérateurs de traitement d'images puissants, rapides et adaptés à l'extraction des structures fines et tortueuses de l'image.

Plus généralement, filtrer les objets non désirés est une étape classique en traitement d'images. La morphologie mathématique est basée sur une approche ensembliste et utilise des éléments structurants pour obtenir de l'information sur la morphologie des objets. Dans [204], Serra et Vincent ont remarqué que souvent, de simples ouvertures et fermetures sont suffisantes pour la majorité des applications. Néanmoins, lorsque les structures à préserver sont des objets fins, les capacités de filtrage sont plus grandes si l'élément structurant s'adapte au contenu de l'image (chapitre 4). Analyser chaque composante connexe permet donc de créer des filtres efficaces, puisqu'ils simplifient l'image sans modifier ses contours. Cela a amené Klein en 1976 [106] à définir les ouvertures binaires par reconstructions qui ont été étendues à des images en niveaux de gris par Vincent [244] en 1993. Dans la même année, Vincent [243] proposa des ouvertures aréolaires, qui préservent ou suppriment les composantes connexes en fonction de leur surface. En 1995, Salembier et Serra [200] filtrèrent les régions plates de l'image et en 1996, Breen et Jones [29] généralisèrent la notion d'ouverture par attributs et proposèrent d'utiliser des critères non croissants pour créer de nouveaux filtres. Seules les propriétés d'anti-extensivité et d'idempotence sont préservées ce qui permet de qualifier ces opérateurs d'amincissements [203, 29]. Ainsi, Wilkinson et Urbach exploitèrent les amincissements en utilisant un critère basé sur l'inertie pour détecter les structures allongées dans l'image [254, 237]. Enfin, Salembier proposa récemment une

règle de filtrage basée sur une méthode d'optimisation globale sous contraintes afin de minimiser le nombre ou le périmètre des zones plates [195].

On reprendra ainsi le travail de Breen and Jones et de Wilkinson et Urbach afin de créer de nouveaux filtres à partir de critères basés sur le diamètre barycentrique et sur l'axe médian de l'objet. Ce chapitre est organisé de la manière suivante : la section 5.2 rappelle les notions de base sur les ouvertures par attributs. Les sections 5.3, 5.4 et 5.5 présentent les nouveaux attributs, puis la section 5.6 associe ces attributs à des amincissements. Enfin, nous donnerons quelques aspects pratiques en étudiant l'algorithme, les différentes optimisations possibles, la complexité ainsi que les temps de calcul en les comparant avec différents opérateurs destinés aux mêmes applications (sections 5.7 et 5.8).

## 5.2 Rappel : amincissement par attributs

Soit  $I : D \rightarrow \{0,1\}$  une image binaire avec  $D$  un sous-ensemble fini, généralement rectangulaire, de  $\mathbb{Z}^2$ . Un opérateur binaire est une fonction qui transforme une image binaire en une autre image binaire. Soit  $X = \{x \in D | I(x) = 1\}$  et  $X^c$  l'ensemble complémentaire dans  $D$ . On associe à  $D$  un voisinage permettant de décrire les liaisons entre les pixels adjacents : on utilise ici une 8-connectivité. Avec cette connectivité, on obtient un ensemble de composantes connexes, écrit  $\{X_i\}$ . Dans le reste de ce chapitre, par *objet* nous comprenons une composante connexe  $X_i$  de l'ensemble  $\{X_i\}$ .

### 5.2.1 Composantes connexes et attributs

Nous souhaitons conserver ou supprimer les objets d'une image avec des attributs intuitifs comme la longueur, la tortuosité, l'élongation ou la circularité. Pour une composante connexe  $X_i$ , ces attributs seront respectivement notés  $Lg(X_i)$ ,  $T(X_i)$ ,  $E(X_i)$  ou  $C(X_i)$  et leurs définitions seront données plus tard. Ces attributs permettent de définir des *critères* comme "plus grand ou égal à" ( $Lg(.) \geq L$ ), ou "moins tortueux que" ( $T(.) < L$ ), avec un paramètre  $L \in \mathbb{R}^+$ . Formellement, un critère  $\chi$  est une fonction qui lie l'ensemble des composantes connexes à vrai ou faux.

Un opérateur binaire, basé sur un critère  $\chi$  est défini par l'équation suivante :

$$\Psi_\chi(X_i) = \begin{cases} X_i & \text{si } \chi(X_i) \text{ est vrai} \\ \emptyset & \text{sinon,} \end{cases} \quad (5.1)$$

pour toutes les composantes connexes  $X_i$  incluses dans  $\{X_i\}$ .

### 5.2.2 Amincissements binaires par attributs

À partir de l'opérateur binaire  $\Psi_\chi$ , l'amincissement binaire par attribut est donnée, pour tout ensemble  $X$ , par :

$$\Phi_\chi(X) = \bigcup_i \Psi_\chi(X_i). \quad (5.2)$$

Ce filtre est donc équivalent à scanner toutes les CCs de l'image puis à les conserver ou à les supprimer entièrement en fonction du critère  $\chi$ .

La transformation duale des amincissements est appelée *épaississement* et elle est simplement définie en calculant un amincissement sur le complémentaire de l'image :

$$\Theta_\chi(X) = [\Phi_\chi(X^c)]^c. \quad (5.3)$$

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSQUES ET GÉOMÉTRIQUES

Dans le reste de ce chapitre, on ne proposera que des amincissements pour la détection de structures claires, puisque les épaisissements peuvent être déduits de ce premier opérateur avec l'équation 5.3.

### 5.2.3 Amincissements en niveaux de gris par attributs

Une image en niveaux de gris  $f$  est une application de  $D$  dans  $V = \{0, \dots, M-1\}$ , avec  $M \in \mathbb{N}$  le nombre de niveaux de gris. L'image  $f$  peut être décomposée en  $M$  images binaires, obtenues par seuillages successifs de  $f$ . On note  $X^h(f)$ , l'image binaire obtenue en seuillant  $f$  avec un seuil  $h \in V$  :  $X^h(f) = \{x \mid x \in D, f(x) \geq h\}$  et  $X_i^h(f)$ , la  $i^{\text{ième}}$  composante connexe de  $X^h(f)$ .

Tout opérateur binaire et croissant,  $\phi$ , comme les ouvertures, peut être généralisé à des images en niveaux de gris en appliquant ce même opérateur à toutes les images binaires extraites de  $f$ ,  $X^h(f)$ , et en recomposant les résultats obtenus pour former à nouveau une image en niveaux de gris :

$$\phi^{\text{directe}} f(x) = \vee \{h \in V \mid x \in \Phi(X^h(f))\}. \quad (5.4)$$

Néanmoins, les attributs que l'on utilise ne sont pas croissants et l'extension à des images en niveaux de gris n'est pas unique. On décrit dans ce document, quatre règles de filtrage tirées de la littérature pour construire cette extension [29, 199, 237] :

1. **Règle Min** : une composante connexe notée  $X_i$  est supprimée si  $\chi(X_i) = 0$  ou s'il existe une composante connexe  $X_j$  telle que  $X_i \subset X_j$ , qui est supprimée.
2. **Règle Max** : une composante connexe,  $X_i$  est supprimée si  $\chi(X_i) = 0$  et si toutes les composantes connexes  $X_j$  telles que  $X_j \subset X_i$ , sont supprimées.
3. **Règle directe** : une composante connexe  $X_i$  est supprimée si  $\chi(X_i) = 0$ . Les autres composantes connexes ne sont pas modifiées. Cette règle est la plus simple de toutes puisqu'elle respecte l'équation 5.4 des ouvertures.
4. **Règle soustractive** : une composante connexe  $X_i$  est supprimée si  $\chi(X_i) = 0$ , cependant, toutes les autres composantes connexes  $X_j$  telles que  $X_j \subset X_i$  sont abaissées par la valeur du contraste de  $X_i$ .

D'autres règles ont été introduites pour réaliser des traitements plus spécifiques qui ne seront pas étudiées dans ce document : la règle médiane [198] réalise un pré-filtrage afin d'améliorer la robustesse de la décision, la règle Viterbi [199] résout un problème d'optimisation pour les mêmes raisons, enfin les règles *k-subtractive* et *k-absorption* [175] fonctionnent avec des filtres hyperconnectés. On remarque que toutes ces extensions fournissent le même résultat si on les applique à des opérateurs croissants comme les ouvertures.

La figure 5.1 présente des amincissements avec un attribut non croissant sur un signal 1-D en niveaux de gris avec différentes règles de filtrage. Les résultats obtenus sont très différents et le choix de cette règle dépend de l'application. Néanmoins, il a été montré dans [237] que seule la règle soustractive (avec les règles *k-subtractive* et *k-absorption* lorsque  $k = 0$ ) satisfait deux propriétés très intuitives : (i) après filtrage, l'image doit être uniquement composée de structures qui respectent le critère  $\chi$ . Et (ii), l'image des différences,  $(f - \phi_\chi(f))$  ne doit contenir que des structures qui ne respectent pas  $\chi$ .

Ces deux propriétés sont très intéressantes puisque l'on obtient, avec un unique filtrage, toutes les structures qui respectent le critère ainsi que celles que ne le respectent pas. De plus, cette règle soustractive permet d'extraire les objets du fond de l'image et facilite donc les tâches de segmentation. Pour ces deux raisons, on utilisera la *règle soustractive* pour nos applications.

Les deux sections suivantes introduisent les attributs géodésiques et géométriques qui seront utilisés dans la suite de ce chapitre pour construire de nouveaux filtres par attributs.

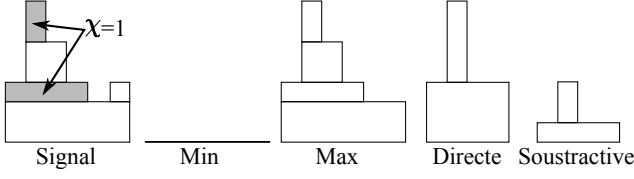


FIGURE 5.1 – Résultat des amincissements par attributs géodésiques avec différentes règles de filtres. Avec respectivement de gauche à droite : le signal initial, avec deux composantes connexes respectant le critère (CC en gris), la règle min, la règle max, la règle directe et la règle soustractive.

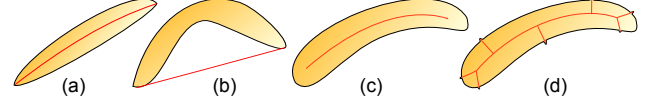


FIGURE 5.2 – Illustration des différentes définitions possibles pour mesurer la longueur d'un objet. (a) et (b) : la longueur du segment entre ses points terminaux. (c) et (d), nombre de points du squelette de l'objet.

### 5.3 Attributs géodésiques

Les attributs géodésiques peuvent être définis dans les domaines discret ou continu. Cependant, on rappelle que les calculs sont réalisés dans  $\mathbb{Z}^2$ , avec une 8-connectivité.

#### 5.3.1 Diamètre géodésique

Lantuéjoul, Beucher et Maisonneuve [115, 116] posèrent une question : “*Comment définir la longueur d'un objet ?*” La première idée consiste à mesurer la longueur du segment connectant ses points terminaux comme le montre les figures 5.2 a et 5.2 b. Néanmoins, cette définition n'est pas satisfaisante puisque ce segment n'est pas toujours inclus dans l'objet. De plus, la notion d'extrémités n'est pas intuitive pour des objets ayant une forme quelconque. Une autre possibilité consiste à calculer des mesures topologiques, comme le cardinal de points du squelette homotopique de l'objet (figures 5.2 c et d). Ces mesures sont bien meilleures puisque le squelette est toujours inclus dans l'objet, néanmoins, de petites modifications sur le contour de l'objet engendrent des variations importantes sur le squelette et donc sur la mesure de longueur qui en découle. Par conséquent, cette mesure de longueur, basée sur le squelette, ne fournit pas de bons résultats et on préférera utiliser les axes médians robustes au bruit comme mesure géométrique (section 5.5). Enfin, on peut utiliser le diamètre géodésique pour obtenir une estimation satisfaisante de la longueur d'un objet.

On résume ici le travail de Lantuéjoul et Maisonneuve [116] puis nous décrivons la généralisation apportée par Soille dans [206]. Soit  $X$  un objet et  $x, y$  deux points appartenant à  $X$ . Il existe une multitude de chemins liant  $y$  à  $x$  qui sont inclus dans  $X$ . Parmi tous ces chemins, ceux qui ont une *longueur géodésique* minimale sont appelés *des arcs géodésiques*. On écrit leur longueur  $d_X(x, y)$  et on visualise ces chemins dans la figure 5.3. Pour un  $x$  donné appartenant à  $X$ , on peut mesurer la longueur des plus grands arcs géodésiques partant de ce point :

$$l_x(X) = \sup_{y \in X} d_X(x, y). \quad (5.5)$$

$l_x(X)$  correspond à la distance du point le plus éloigné de  $x$ , avec une distance géodésique. Dans l'illustration de la figure 5.3 d, on attribue à tous les pixels  $x \in X$  de l'objet, la valeur de la distance  $l_x(X)$ . Les points où  $l_x(X)$  atteint un minimum global sont appelés *des centres géodésiques* de  $X$ . Au contraire, les points de  $X$  où  $l_x(X)$  est maximal, sont des *extrémités géodésiques* de  $X$ .

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSQUES ET GÉOMÉTRIQUES

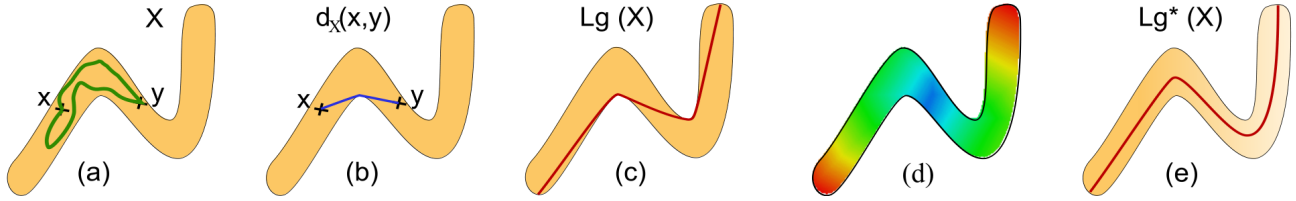


FIGURE 5.3 – Illustration de la distance géodésique. (a) deux chemins arbitraires entre  $x$  et  $y$ . (b) arc géodésique entre ces deux points. (c) le plus grand arc géodésique entre tous les couples de points de  $X$ , sa longueur est appelée diamètre géodésique. (d) pour tout  $x \in X$ , illustration de la valeur de la distance  $l_x(X)$  en fausses couleurs (du bleu au rouge, de la plus faible valeur à la plus forte). (e) diamètre géodésique utilisant une distance géodésique généralisée [206].

Le *diamètre géodésique* de  $X$  est défini comme la longueur du plus grand arc géodésique (voir la figure 5.3c) :

$$Lg(X) = \sup_{x \in X} l_x(X), \quad (5.6)$$

ce qui correspond à la plus grande distance géodésique entre deux points appartenant à  $X$  :

$$Lg(X) = \sup_{x,y \in X} d_X(x,y). \quad (5.7)$$

Le diamètre géodésique est de notre point de vue, une bonne estimation de la longueur d'un objet. Il a, de plus, trois avantages [116] :

1. C'est une définition générale, valide pour tous les objets quelles que soient leur forme et tortuosité.
2. Le diamètre géodésique est robuste au bruit, puisque des petits changements dans la forme de l'objet (en excluant tout changement de topologie) vont causer, tout au plus, un petit changement dans cette mesure.
3. Le calcul de  $Lg(X)$  fournit par la même occasion d'autres attributs comme l'élongation, la tortuosité et la circularité.

Soille proposa dans [206], une généralisation de la distance géodésique. Cette extension consiste à calculer une carte de distance Euclidienne dans l'objet, à l'inverser puis à réaliser une propagation avec la notion de temps géodésique [207]. Ainsi, le diamètre géodésique obtenu suit l'axe médian de l'objet, comme le montre la figure 5.3(e). Ceci augmente la précision et la robustesse de la mesure du diamètre géodésique puisque une fibre aura la même mesure avant et après un changement de tortuosité. On notera, dans la suite de ce chapitre, tous les attributs basés sur cette distance géodésique généralisée par une étoile \*. On remarque que dans la pratique, les résultats sont très similaires pour la détection de structures fines. Cependant, les temps sont au minimum doublés par rapport aux distances géodésiques classiques. On utilisera alors cette distance géodésique généralisée que lorsque la précision de l'opérateur doit être importante (on prend l'exemple du calcul des chemins d'écoulement [211] ou l'interpolation des données de contour [80]). Pour obtenir un opérateur le plus rapide possible, on utilisera la distance géodésique classique.

On remarque aussi que la distance géodésique et sa généralisation, sont définies de la même manière dans un espace à  $n$ -dimensions. Ainsi, la généralisation de ces opérateurs à des espaces de dimension supérieure est directe.

### 5.3.2 Elongation géodésique

Le diamètre géodésique fournit une définition satisfaisante pour mesurer la longueur d'un objet. Cependant, il ne donne que peu d'information sur sa forme. En combinant le diamètre géodésique avec la surface occupée par  $X$  (notée  $S(X)$ ), on obtient des informations sur l'élongation de l'objet. L'élongation géodésique est une notion introduite par Lantuéjoul et Maisonneuve dans [116] que l'on calcule par la relation suivante :

$$E(X) = \frac{\pi Lg^2(X)}{4S(X)}. \quad (5.8)$$

Plus un objet est long et fin, et plus son élongation sera importante. La borne minimale est atteinte avec un disque puisque le facteur de normalisation est réglé pour obtenir un allongement de 1 pour cet objet.

En dimension trois, Parra-Denis [177] a étendu cet opérateur pour que le paramètre de normalisation soit égal à 1 pour une sphère :

$$E(X) = \frac{\pi Lg^3(X)}{6V(X)}, \quad (5.9)$$

avec  $V(X)$  le volume de la composante connexe.

### 5.3.3 Un nouvel attribut : la tortuosité géodésique

On propose un nouvel attribut, dérivé du diamètre géodésique, la tortuosité géodésique. Toute paire de points  $\{x, y\}$  est une paire d'extrémités géodésiques de  $X$ , si et seulement si  $d_X(x, y) = Lg(X)$ . Cependant, on remarque qu'un objet peut avoir plusieurs paires d'extrémités géodésiques (un disque par exemple) et cet ensemble est noté  $\Xi(X) = \{\{x_0, y_0\}, \{x_1, y_1\}, \dots\}$ . On définit alors  $L_{Eucl}(X)$ , comme la distance euclidienne minimale entre les extrémités géodésiques :

$$L_{Eucl}(X) = \min_{\{x, y\} \in \Xi(X)} \|x, y\|. \quad (5.10)$$

Ainsi, la tortuosité géodésique correspond au ratio entre le diamètre géodésique et  $L_{Eucl}(X)$  :

$$T(X) = \frac{Lg(X)}{L_{Eucl}(X)}. \quad (5.11)$$

Par conséquent, ce paramètre prend des valeurs supérieures à 1 et tous les objets convexes auront une tortuosité de 1.

### 5.3.4 Circularité géodésique

Calculer la circularité d'un objet, revient à simplement calculer l'inverse de l'élongation :

$$C(X) = \frac{1}{E(X)}. \quad (5.12)$$

Cet attribut prend donc des valeurs comprises entre  $]0, 1]$  et permet de visualiser si la forme d'un objet est proche d'un cercle ou non. Un exemple d'utilisation de cet attribut est présenté dans la figure 5.11.

Dans l'espace Euclidien, tous ces attributs sont invariants par rotation et, pour les attributs d'élongation, de circularité et de tortuosité, ils sont de plus invariants par changement d'échelle. Néanmoins,



## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSIQUES ET GÉOMÉTRIQUES

si l'objectif de cette détection est par exemple de discriminer les structures longues et allongées des structures courtes et allongées, on peut combiner plusieurs attributs entre eux pour être plus restrictif :

$$\begin{aligned} LgT(X) &= Lg(X) \times T(X), \\ LgE(X) &= Lg(X) \times E(X), \\ LgC(X) &= Lg(X) \times C(X), \\ &\dots \end{aligned} \tag{5.13}$$

Tous ces attributs sont basés sur le diamètre géodésique et le calcul de cette mesure nécessite un temps considérable. Ainsi, pour obtenir des algorithmes rapides, on définit un nouvel attribut que l'on appelle *le diamètre barycentrique*. Il a pour objectif d'offrir une approximation très rapide du diamètre géodésique. On présente cet attribut dans la section suivante 5.4, en étudiant précisément l'erreur commise par cette approximation.

### 5.4 Un nouvel attribut géodésique : le diamètre barycentrique

L'objectif de cet attribut est de remplacer le diamètre géodésique par une mesure similaire et ayant une complexité algorithmique plus faible. Nous décrivons premièrement les algorithmes permettant de calculer le diamètre géodésique présent dans la littérature, puis nous introduirons ce nouvel attribut.

#### 5.4.1 Description des algorithmes pour le calcul du diamètre géodésique

L'approche directe, pour le calcul du diamètre géodésique d'un objet  $X$ , consiste à réaliser des *propagations*. Une propagation consiste à calculer, à partir d'un pixel  $x$  de l'objet,  $l_x(X)$ . En réalisant une propagation à partir de tous les pixels du bord de l'objet, on pourra alors obtenir la valeur du diamètre géodésique. La complexité de cet algorithme est élevée et dépend à la fois de la surface ainsi que du nombre de pixels du bord de l'objet. On remarque qu'il y a de nombreuses propagations inutiles amenant à une implémentation inefficace. Schmitt [201] prouva qu'il était suffisant de considérer seulement un sous ensemble des pixels du bord comme point de départ pour une propagation. Malgré l'apport considérable de cet algorithme, il reste néanmoins trop de propagations superflues.

Lantuéjoul et Maisonneuve ont créé un algorithme parallèle pour le calcul du diamètre géodésique sur une grille hexagonale [116]. Soit  $X$  une composante simplement connexe et  $Y$  l'ensemble des points du bord de l'objet. En utilisant une propagation particulière dans la grille hexagonale et à partir des points de  $Y$ , l'algorithme fournit le résultat en une unique propagation. Cependant, cet algorithme ne peut traiter que des composantes connexes simplement connectés, puisque le front de propagation tournera infiniment autour de ses trous. Cette caractéristique rend cet algorithme inadéquat pour notre application.

Parra-Denis, dans [177], commence par calculer la fonction distance dans l'objet, puis réalise des dilatations conditionnelles à  $X$  afin de repérer les points maxima qui sont des extrémités de l'objet. Enfin, à partir de chacune de ces extrémités, une propagation géodésique est réalisée pour obtenir le diamètre géodésique. Avec cet algorithme, la complexité est élevée :  $\mathcal{O}(n^2)$  avec  $n$  le nombre de pixels de la composante connexe. L'analyse de la complexité sera donnée plus en détail dans la section 5.8.2.

Ces méthodes ne nous permettent pas de calculer rapidement le diamètre géodésique et nous proposons une méthode très efficace pour réaliser un calcul approché de cet attribut : *le diamètre barycentrique*.

### 5.4.2 Distance géodésique itérée

Pour tous les points de  $x \in X$ , pouvons nous calculer la longueur  $l_x(X)$  du plus grand arc géodésique en une seule propagation en partant de  $x$  ? Est ce que cette valeur peut remplacer  $Lg(X)$  ? Est-ce une bonne approximation ? Comment choisir le point de départ  $x$  ?

Considérons l'arc géodésique maximal de  $X$ . Soient  $y_1$  et  $y_2$  ses extrémités. La distance géodésique entre ces deux points est égale à  $Lg(X) = d_X(y_1, y_2)$ . Etant donné que la distance géodésique est une distance [116], l'inégalité triangulaire s'applique et on a pour tout  $x \in X$  :

$$d_X(y_1, y_2) \leq d_X(y_1, x) + d_X(x, y_2). \quad (5.14)$$

De plus, par définition,  $l_x(X)$  est à la fois plus grand ou égal à  $d_X(y_1, x)$  et à  $d_X(x, y_2)$  tout en étant plus petit que  $Lg(X)$ . On obtient :

$$Lg(X) \geq l_x(X) \geq \frac{Lg(X)}{2}. \quad (5.15)$$

Aussi, l'erreur relative  $(Lg(X) - l_x(X))/Lg(X)$  que l'on commet lorsque l'on approxime  $Lg(X)$  par  $l_x(X)$  est toujours inférieure à 50%. Pour améliorer cette approximation, nous pouvons réaliser d'autres propagations en partant cette fois ci d'un des points les plus éloignés de  $x$  avec la distance géodésique. Soit  $Y \subset X$ , cet ensemble de points :

$$Y = \{y \mid y \in X, d_X(x, y) = l_x(X)\}. \quad (5.16)$$

On peut donc introduire la distance géodésique maximale itérée partant du point initial  $x$  :

$$l_x^2(X) = \sup_{y \in Y} l_y(X). \quad (5.17)$$

On peut montrer facilement que :

$$Lg(X) \geq l_x^2(X) \geq l_x(X) \geq \frac{Lg(X)}{2}, \quad (5.18)$$

et que plus généralement  $l_x^n(X)$  converge,  $\forall n \in \mathbb{Z}^+$  (avec  $n$  est le nombre d'itérations) :

$$Lg(X) \geq l_x^n(X) \geq l_x^{n-1}(X) \geq \frac{Lg(X)}{2}. \quad (5.19)$$

Néanmoins, la limite de cette suite n'est pas forcément égale à  $Lg(X)$  mais on note qu'en pratique, la convergence est très rapide et qu'aller au delà de  $l_x^2(X)$  n'est pas intéressant. Sur ce point précis, une analyse plus précise est donnée à la fin de cette section. La figure 5.4 fournit un exemple où  $l_x^2(X)$  est une approximation bien plus précise du diamètre géodésique que  $l_x(X)$  et où d'autres itérations n'augmentent pas la précision de cette approximation.

Dans les tests qui suivent, on utilisera  $l_x^2(X)$  comme approximation du diamètre géodésique. Il reste cependant à choisir  $x$ , le point initial de la première propagation.

### 5.4.3 Choix du point de départ

Le choix du point  $x \in X$  a une influence sur la qualité de l'approximation obtenue. Pour le montrer, nous avons mis en place une série de tests, et plusieurs stratégies ont été testées.

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSIQUES ET GÉOMÉTRIQUES

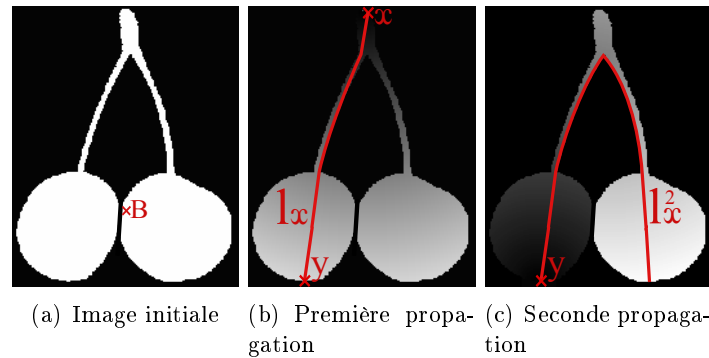


FIGURE 5.4 – Illustration de l’apport de  $l_x^2(X)$  par rapport à  $l_x^1(X)$  sur un objet  $X$  quelconque. Le point B de l’image (a) est le barycentre de la composante connexe et le point de départ,  $x$ , correspond au point le plus éloigné de B avec une distance euclidienne. La première propagation donne  $l_x$ , suivie de la seconde propagation en partant de  $y$ , qui fournit une bien meilleure approximation du diamètre géodésique :  $l_x^2$ .

## Description des différentes stratégies

Le choix de  $x$  peut être réalisé de plusieurs façons différentes :

- Lorsque  $x$  est choisi comme le point le plus éloigné du barycentre de l'objet  $X$ , on obtient ce que l'on appelle le *diamètre barycentrique*, noté  $Lg_{BarFar}(X)$ . On remarque que le barycentre de cet objet n'appartient pas forcément à  $X$  et par conséquent, on utilise la distance euclidienne pour trouver les points les plus éloignés du barycentre. En partant d'un de ces points, on suppose qu'il sera placé près de l'une des extrémités de  $X$  (ce ne sera pas forcément le cas pour des objets non convexes).
- De la même manière, au lieu de choisir le point le plus éloigné du barycentre, on peut choisir le point le plus proche (avec toujours une distance euclidienne). On obtient une autre approximation de  $L(X)$  :  $Lg_{BarNear}(X)$ .
- Le barycentre n'est pas forcément inclus dans l'objet et pour remédier à cela, on peut utiliser un des points correspondant au centre géodésique. Cependant, trouver cet ensemble de points est tout aussi complexe que la recherche du diamètre géodésique. Par conséquent, on calcule une carte de distance à l'intérieur de l'objet pour prendre le pixel le plus éloigné d'un des bords. On obtient respectivement deux mesures associées à ce point  $Lg_{CentreNear}$  et  $Lg_{CentreFar}$  (en choisissant le point le plus proche ou le plus éloigné).
- Enfin, le point de départ  $x$  sera le premier pixel de l'objet trouvé avec un balayage ligne par ligne de l'image. On note la mesure associée  $Lg_{Random}$ .

Ces cinq mesures peuvent donc être utilisées pour calculer une approximation du diamètre géodésique avec seulement deux propagations. Pour ces stratégies, on calcule les erreurs relatives moyennes sur de nombreux objets binaires répartis en six bases de données.

## Description des bases de données

Pour avoir une bonne estimation des erreurs moyennes pour chaque méthode, il est nécessaire d'avoir un nombre important d'objets à notre disposition. On utilise donc six bases de données différentes dont cinq sont constituées d'objets générés aléatoirement :

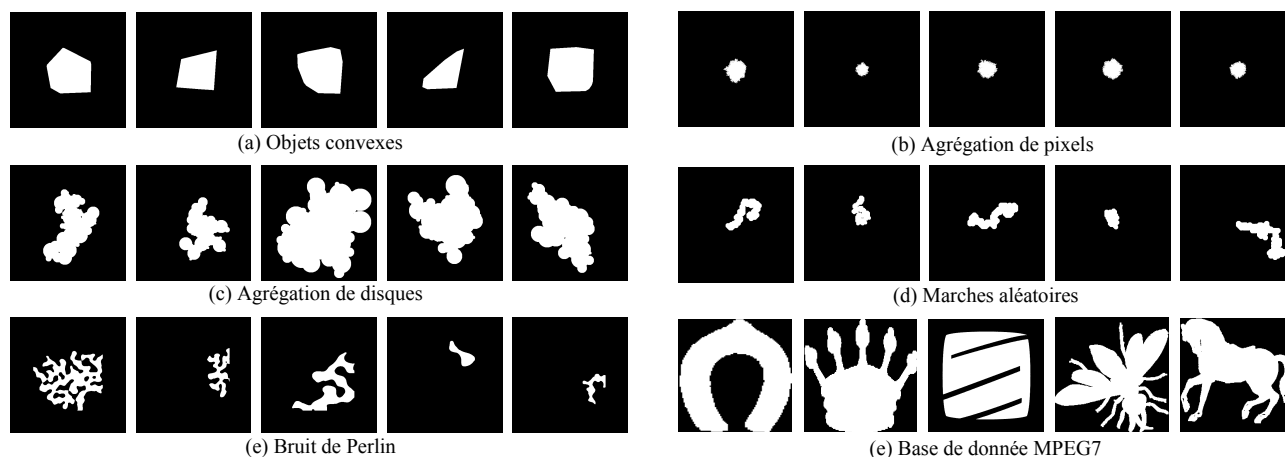


FIGURE 5.5 – Illustration de cinq objets binaires pour chaque méthode.

- **Objets convexes** : Un nombre aléatoire de points est généré (entre 10 et 100) puis ils sont placés aléatoirement dans l'espace. L'enveloppe convexe de ces points forme un objet convexe.
  - **Agrégation de pixels** : Cette méthode est utilisée pour générer des objets denses et irréguliers sur les bords. L'ensemble est initialisé à un unique point et à chaque itération, on ajoute aléatoirement un des pixels appartenant au voisinage de cette forme. On répète ce processus un nombre aléatoire de fois.
  - **Agrégation de disques** : Cette méthode fonctionne comme la méthode *Agrégation de pixels* sauf qu'à chaque itération, on ajoute un disque ayant un rayon aléatoire (compris entre 5 et 40 pixels) au lieu d'un unique pixel. On obtient des objets complexes.
  - **Marches aléatoires** : on commence par choisir la taille d'un disque que l'on place dans l'image. Puis nous utilisons un algorithme des marches aléatoires pour simuler un mouvement Brownien et un objet complexe.
  - **Bruit de Perlin** : Le bruit de Perlin [182] permet de construire une texture procédurale dont l'apparence est facilement contrôlable et multi-échelles. En seuillant cette texture et en conservant la plus grande composante connexe de l'image, on obtient des objets très tortueux.
  - **Base de données MPEG7**<sup>1</sup> : cette base de données classique regroupe 1400 objets binaires.
- Nous générons 10000 objets binaires par méthode, pour obtenir un total de 51400 objets. Pour illustrer la forme et la variabilité de ces objets, cinq réalisations sont présentées et regroupées dans la figure 5.5.

## Résultats

Sur l'ensemble de ces objets binaires, nous mesurons le diamètre géodésique ainsi que toutes les approximations afin de calculer l'erreur relative moyenne pour chaque méthode, l'erreur maximale ainsi que l'écart type. Le tableau 5.1 regroupe tous les résultats de cette étude et on remarque tout d'abord que les erreurs relatives moyennes sont, en pratique, très éloignées de la limite théorique des 50%. En effet, les erreurs moyennes sont toutes inférieures à 2% et les erreurs maximales à 30%. De plus, on constate que le diamètre barycentrique propose les meilleurs résultats en réduisant de 20% (au

1. Base de données disponible sur [www.imageprocessingplace.com/rootfilesV3/imagetdatabases.htm](http://www.imageprocessingplace.com/rootfilesV3/imagetdatabases.htm)

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSIQUES ET GÉOMÉTRIQUES

TABLEAU 5.1 – Tableau récapitulatif des erreurs relatives entre la mesure du diamètre géodésique et les différentes approximations.

Méthodes		$Lg_{BarFar}$	$Lg_{BarNear}$	$Lg_{CentreFar}$	$Lg_{CentreNear}$	$Lg_{Random}$
Objets convexes	Mean (%)	<b>0,24</b>	<b>0,24</b>	0,47	0,56	1,92
	Std (%)	0,85	<b>0,84</b>	1,41	1,53	3,36
	Max (%)	<b>10,04</b>	11,20	15,9	17,24	22,59
Agrégation de pixels	Mean (%)	0,43	<b>0,40</b>	0,74	0,85	2,68
	Std (%)	1,09	<b>1,03</b>	1,58	1,71	3,59
	Max (%)	10,76	<b>9,27</b>	12,87	12,87	24,59
Agrégation de disques	Mean (%)	<b>0,22</b>	<b>0,22</b>	0,23	0,59	1,21
	Std (%)	<b>0,72</b>	1,26	1,52	2,31	4,53
	Max (%)	<b>7,17</b>	12,75	13,84	16,85	28,83
Marches aléatoires	Mean (%)	<b>0,13</b>	0,28	0,24	0,53	1,31
	Std (%)	<b>0,66</b>	1,03	1,08	1,61	3,27
	Max (%)	<b>9,97</b>	21,7	21,48	21,48	29,96
Bruit de Perlin	Mean (%)	<b>0,12</b>	0,16	0,21	0,40	0,76
	Std (%)	<b>0,74</b>	0,80	1,21	1,50	2,61
	Max (%)	10,87	<b>10,14</b>	16,55	13,61	31,90
Données MPEG7	Mean (%)	<b>0,25</b>	0,43	0,34	0,58	0,66
	Std (%)	<b>1,12</b>	1,93	1,44	1,91	2,38
	Max (%)	20,36	22,31	<b>19,16</b>	19,21	30,11
Tous les objets	Mean (%)	<b>0,23</b>	0,29	0,37	0,58	1,42
	Std (%)	<b>0,86</b>	1,49	1,37	1,76	3,29
	Max (%)	<b>20,36</b>	22,31	21,48	21,48	31,90

minimum) l'erreur relative moyenne par rapport aux autres approximations. **Par conséquent, on choisira  $Lg_{BarFar}$  comme approximation du diamètre géodésique.**

Pour cette mesure, l'erreur maximale n'est que de 20% sur ces 51400 objets, loin de la limite théorique. Il est donc légitime de se demander si cette limite peut être atteinte. La figure 5.6 présente un exemple où  $Lg_{BarFar}(X)$  est très proche de  $\frac{Lg(X)}{2}$  et où  $l_x^n$  ne converge pas vers  $Lg(X)$ . En effet,  $Lg_{BarFar}(X)$  est atteint après une unique propagation et itérer le processus n'améliore pas cette mesure.

Pour identifier la vitesse de convergence  $l_x^n$  des différentes approximations en fonction du nombre de propagations  $n$ , on utilise tous les objets de la base de données MPEG7 et on calcule l'erreur relative moyenne entre le diamètre géodésique et les différentes approximations en itérant jusqu'à  $l_x^6$ . Les résultats sont présentés dans la figure 5.7 et une fois de plus, le diamètre barycentrique fournit les meilleurs résultats puisque l'erreur commise est la plus faible pour tous les  $n$ . De plus, si on souhaite avoir une approximation encore plus rapide, réaliser une unique propagation ne pénalise pas trop le résultat qui reste inférieur à 3%. Enfin, on remarque que le passage d'une à deux propagations améliore considérablement les mesures mais au delà de  $n = 2$ , l'apport est minime.

D'un point de vue vitesse de calcul,  $Lg_{random}$  est la mesure la plus rapide mais c'est aussi la plus imprécise. Ensuite viennent les méthodes basées sur le barycentre de l'objet  $Lg_{BarFar}$  et  $Lg_{BarNear}$  et enfin, celles basées sur les maxima de la fonction distance  $Lg_{CentreFar}$  et  $Lg_{CentreNear}$ . Pour conclure

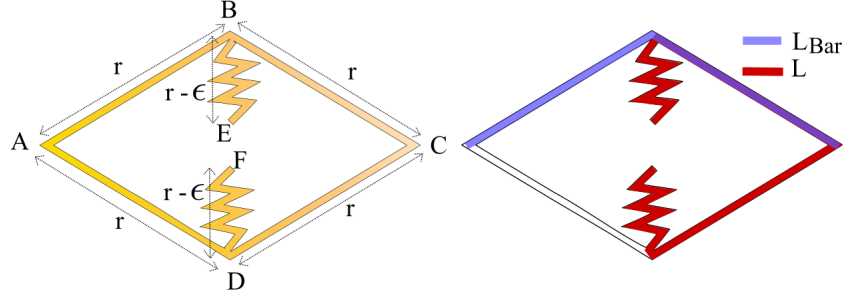


FIGURE 5.6 – Exemple d'un objet  $X$  où le diamètre barycentrique ne converge pas vers le diamètre géodésique. On a ici  $Lg(X) = d_X(E, B) + d_X(B, C) + d_X(C, D) + d_X(D, F) = 4r - 2\epsilon$ , et  $Lg_{BarFar} = d_X(A, B) + d_X(B, C) = 2r$ . L'erreur relative est donc égale à  $\frac{Lg - Lg_{Bar}}{L} = \frac{r - \epsilon}{2r - \epsilon}$  et peut donc être arbitrairement proche de 0,5.

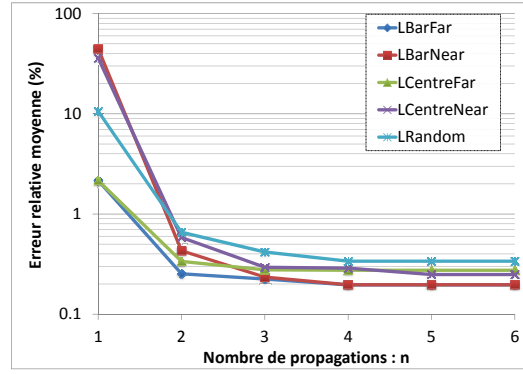


FIGURE 5.7 – Erreur relative en pourcentage et en échelle logarithmique entre la valeur du diamètre barycentrique et celle du diamètre géodésique en fonction du nombre de propagations.

cette section, le diamètre barycentrique  $Lg_{Bar} = Lg_{BarFar}$  est une approximation rapide et précise du diamètre géodésique. Pour la suite de ce document, nous utiliserons  $Lg_{Bar}$  pour calculer la longueur, l'élongation, la tortuosité ou la circularité d'un objet. Nous pouvons aussi définir son pendant en utilisant la distance géodésique généralisée :  $Lg_{Bar}^*$ . Dans la section suivante, nous introduisons des attributs basés sur des mesures géométriques.

## 5.5 Attributs géométriques

Le diamètre géodésique et, par conséquent, le diamètre barycentrique fournissent une bonne représentation de la longueur d'un objet. Cependant, lorsque plusieurs structures filiformes se croisent pour former un réseau, le diamètre géodésique ne réalisera qu'une mesure partielle de ce réseau. Pour pallier à ce problème, nous proposons dans la suite une mesure de la longueur d'un objet basée sur l'axe médian.

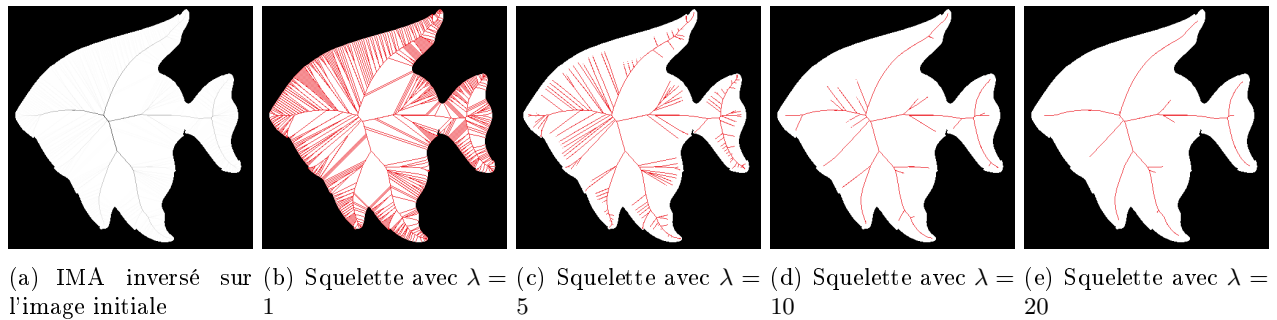


FIGURE 5.8 – Illustration de la méthode *Integer medial axis*. Les branches générées par les discontinuités des frontières sont supprimées en filtrant l'axe médian.

### 5.5.1 Axe médian

En 1967, Blum [22] a introduit les notions d'axe médian et de squelette d'une forme binaire. L'axe médian a été illustré initialement par une propagation d'un feu de forêt à partir de tous les points frontières de l'objet. L'axe médian est donc constitué de points où les fronts de propagations se rejoignent. D'un point de vue géométrique, l'axe médian correspond aux points de l'objet équidistants d'au moins deux points situés sur la frontière de cet objet.

On dénombre deux difficultés principales pour le calcul de cette mesure : (i) le calcul de l'axe médian sur une trame discrète n'est pas aisé, et (ii) il présente une forte sensibilité à des petites perturbations sur le contour de l'objet. Aussi, pour limiter l'impact du bruit sur les frontières de l'objet, de nombreuses méthodes proposent de réaliser une étape de filtrage (voir [7] pour un état de l'art récent). À partir de ces méthodes, on extrait les  $\lambda$ -medial axis [35] qui permettent d'obtenir un axe médian stable et robuste au bruit. D'autres chercheurs reprennent cette notion [129, 93, 34] et on utilisera le travail de Hesselink et Roerdink [93] dénommé *Integer Medial Axis* qui permet de calculer un axe médian précisément (distance euclidienne) et rapidement (en temps constant par pixel). Ainsi, toutes les branches de l'axe médian seront pondérées par la distance entre les deux points frontières qui l'ont généré. Si ces deux points sont très proches, ce qui est généralement le cas lorsque les frontières de l'objet ne sont pas lisses, les branches auront un niveau de gris faible et l'on pourra facilement les filtrer avec un simple seuillage au niveau  $\lambda$ . Cette transformation est notée  $\text{IMA}(X)$  et la figure 5.8 présente le résultat de cette transformation sur une forme binaire pour différentes valeurs de  $\lambda$  pour le filtrage.

### 5.5.2 Attributs basés sur l'axe médian

À partir de l'axe médian en niveaux de gris, il est important de trouver un indicateur de la longueur d'un objet qui permet d'avoir une bonne estimation de sa longueur. Si on utilise le nombre de points du squelette après le filtrage de l'axe médian avec un seuil  $\lambda$  fixe,  $\text{Card}\{\text{IMA}(X) > \lambda\}$ , alors ce filtrage n'est pas invariant par échelle. En effet, le filtrage sera plus ou moins important selon la taille de l'objet. La première ligne de la figure 5.9 permet de visualiser ce phénomène sur trois objets identiques de taille différentes (images  $128 \times 128$ ,  $256 \times 256$ ,  $512 \times 512$  et  $1024 \times 1024$  pixels).

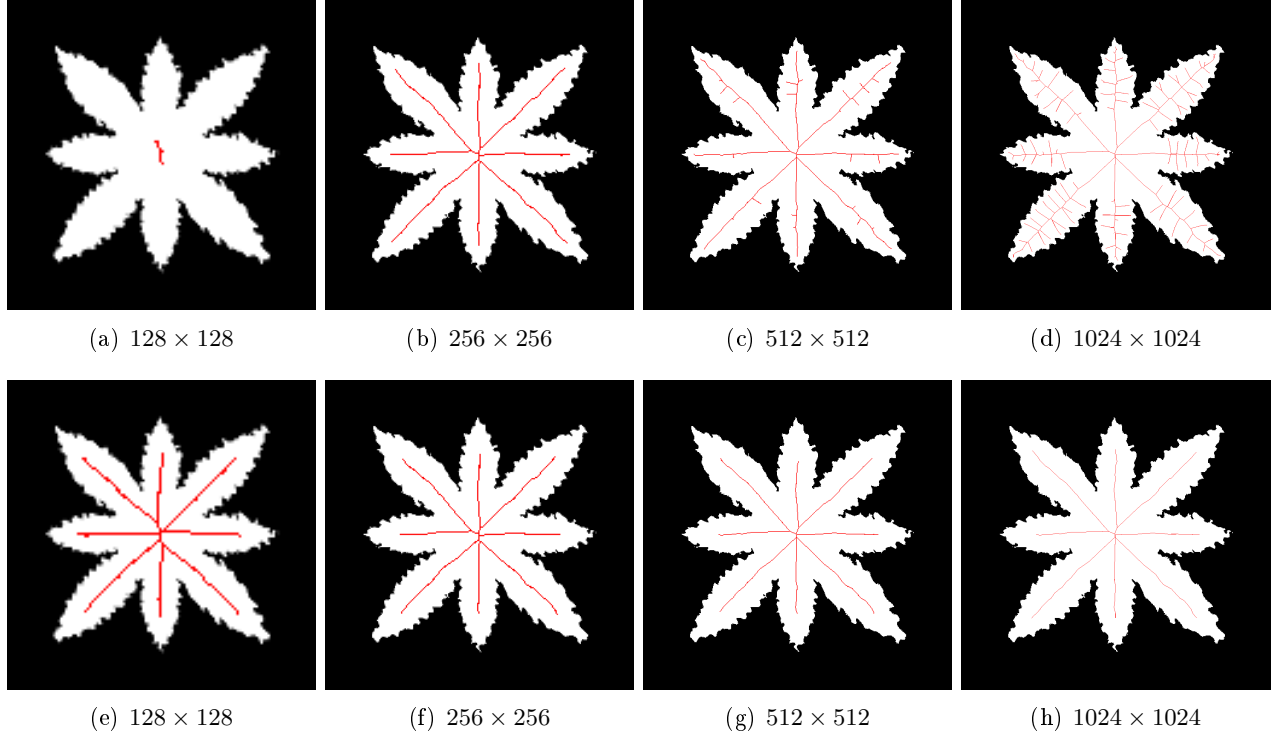


FIGURE 5.9 – Filtrage de l'axe médian : la première ligne présente des résultats des axes médians avec un seuil global fixe ( $\lambda = 20$ ). La seconde ligne présente les résultats avec un seuil adaptatif  $\lambda' = 3$ .

Au lieu d'un seuillage global de l'axe médian, on propose un filtrage adaptatif qui prend en compte la valeur maximale de l'axe médian :

$$L_{\text{IMA}}(X) = \text{Card} \left\{ \text{IMA}(X) > \frac{\max(\text{IMA}(X))}{\lambda'} \right\}. \quad (5.20)$$

Ainsi le filtrage est complètement indépendant de la taille de la structure et permet d'avoir une mesure plus précise de la longueur de l'objet. La seconde ligne de la figure 5.9 illustre cette mesure adaptative avec  $\lambda' = 3$ . Dans le reste de ce chapitre, on fixe la valeur de ce paramètre à 3.

Cette mesure représente uniquement la mesure de la longueur d'un objet. Néanmoins, lorsque  $X$  est un disque, alors l'axe médian correspond au seul centre de l'objet et tous les disques auront une longueur unitaire. Ceci constitue un problème pour calculer l'indice d'élongation et de circularité. Ce problème est résolu en prenant pour l'élongation, le maximum entre 1 et le rapport  $L_{\text{IMA}}(X)^2/S(X)$  :

$$E_{\text{IMA}}(X) = \max \left\{ 1, \frac{L_{\text{IMA}}(X)^2}{S(X)} \right\}, \quad (5.21)$$

et pour l'indice de circularité :

$$C_{\text{IMA}}(X) = \frac{1}{E_{\text{IMA}}(X)}. \quad (5.22)$$

Ces attributs formés à partir de l'axe médian permettent de mesurer plus précisément la forme des réseaux de fibres tout en étant robuste aux déformations des contours de l'objet (voir la figure 5.9).



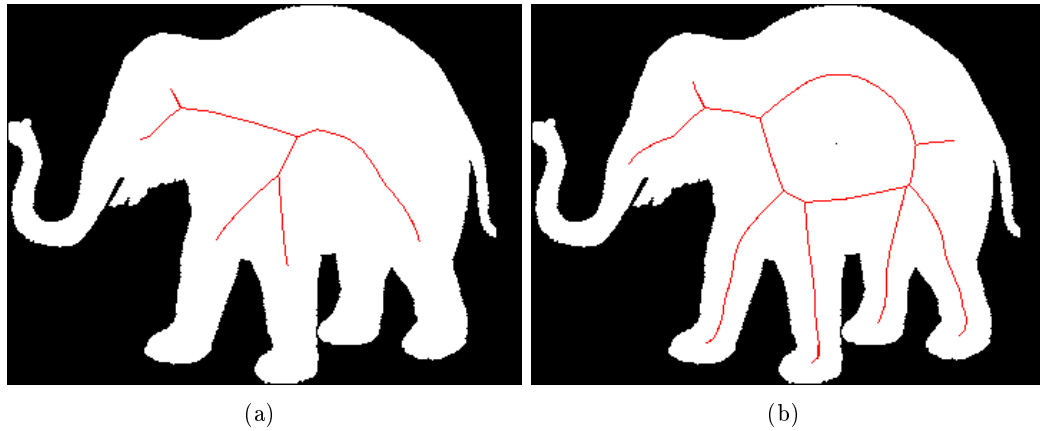


FIGURE 5.10 – Sensibilité aux changements de topologie. Dans (b), un pixel noir a été ajouté au centre de l'objet entraînant une sur-estimation de la longueur de cet objet par rapport à (a), la version originale.

Bien que cet opérateur soit robuste aux déformations des frontières, il reste cependant sensible aux bruits induisant un changement de topologie. En effet, la figure 5.10 présente deux objets identiques où un pixel bruité a été ajouté au centre de l'objet. Puisque la topologie de l'objet a été modifiée, on remarque que l'on obtient deux estimations de longueur très différentes. Pour limiter ce problème, on pourrait supprimer de l'image, tous les trous de taille inférieure à seuil donnée avant de calculer l'axe médian.

Au final, nous disposons de 11 attributs géodésiques ou géométriques répartis en trois catégories. Chaque méthode a des avantages et des inconvénients que l'on résume succinctement ici :

- Quatre attributs sont formés à partir du diamètre barycentrique en utilisant une distance géodésique classique ( $Lg_{Bar}$ ,  $E_{Bar}$ ,  $T_{Bar}$  et  $C_{Bar}$ ). Les temps de calcul sont très rapides et la précision de leur mesure est bonne.
- Quatre autres attributs utilisent une distance géodésique généralisée avec le diamètre barycentrique ( $Lg_{Bar}^*$ ,  $E_{Bar}^*$ ,  $T_{Bar}^*$  et  $C_{Bar}^*$ ). Leur précision est très bonne cependant, les temps de calcul restent élevés.
- Enfin, trois attributs sont formés à partir de l'axe médian d'un objet ( $L_{IMA}$ ,  $E_{IMA}$  et  $T_{IMA}$ ). Ces mesures permettent de mesurer plus précisément la longueur totale des réseaux de fibres mais les temps de calcul sont élevés et cette mesure est sensible aux bruits qui introduisent un changement de topologie.

Dans la section suivante, on définit des nouveaux amincissements avec ces attributs géodésiques et géométriques.

### 5.6 Amincissement par attributs géodésiques et géométriques

À notre connaissance, c'est la première fois que les attributs géodésiques et géométriques, définis dans les sections 5.3, 5.4 et 5.5, sont utilisés pour construire des amincissements par attributs.

On présente ainsi les résultats de ces amincissements, premièrement sur des images binaires puis sur des images en niveaux de gris.

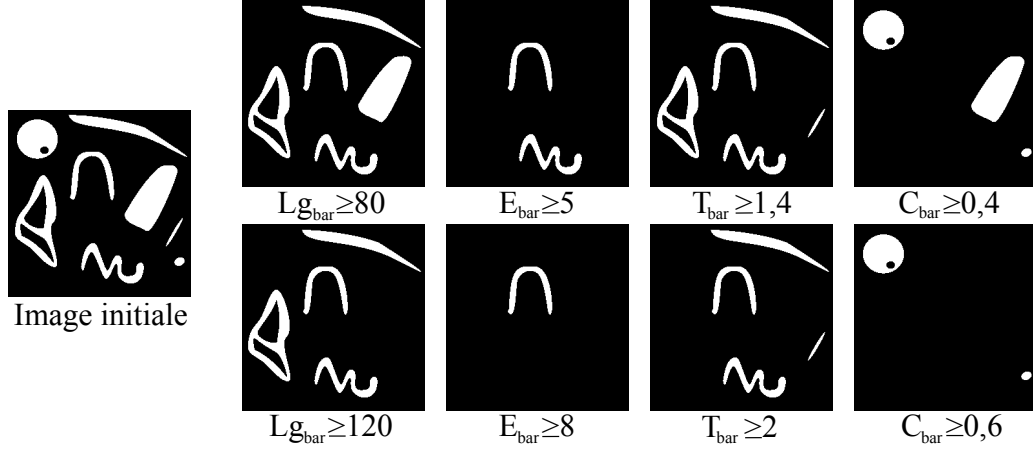


FIGURE 5.11 – Résultat des amincissements avec des attributs géodésiques sur une image binaire. De gauche à droite, l'image initiale, puis l'utilisation de critères de taille, d'allongement, de tortuosité et de circularité.

### 5.6.1 Résultats sur des images binaires

La figure 5.11 présente les résultats des amincissements par attributs géodésiques sur une image test. Cette image regroupe des structures qui sont plus au moins longues, plus ou moins allongées et plus ou moins tortueuses. On souhaite discriminer ces différentes structures en utilisant différents critères géodésiques :

- Suppression de tous les objets dont le diamètre barycentrique est plus petit que 80 pixels ou plus petit que 120 pixels (seconde colonne de la figure 5.11).
- Suppression des objets qui ont un indice d'allongement inférieur à 5 et à 8 (troisième colonne).
- Suppression des objets non tortueux, qui ont un indice de tortuosité inférieur à 1,4 puis à 1,8 (quatrième colonne).
- Enfin, suppression des objets ayant un indice de circularité inférieur à 0,4 puis à 0,6 (cinquième colonne).

On peut observer, à travers cet exemple simple, que les attributs géodésiques permettent de facilement faire la distinction entre les différents objets de l'image. De plus, le choix du critère reste très intuitif et il est ainsi facile de paramétrer ces filtres.

### 5.6.2 Résultats sur des images en niveaux de gris

Dans la section 5.2.3, nous avons choisi la règle soustractive pour étendre ces opérateurs à des images en niveaux de gris. Nous présentons donc quelques exemples de fonctionnement de ces filtres par attributs sur une image d'empreintes digitales avec cette règle soustractive. Les structures sont longues et tortueuses et ces filtres rehaussent certaines structures de l'image.

La figure 5.12 regroupe les 11 attributs que nous avons précédemment définis et nous comparons les résultats obtenus. Avec un critère de longueur et pour un seuil donné de  $L = 600$  pixels,  $Lg_{Bar}$  supprime davantage de structures dans l'image. En effet, le diamètre barycentrique ne suit pas l'axe médian des structures mais il minimise le chemin parcouru en suivant les frontières de l'objet (avec une distance géodésique classique). Cet attribut fournit donc des mesures de longueurs plus courtes pour

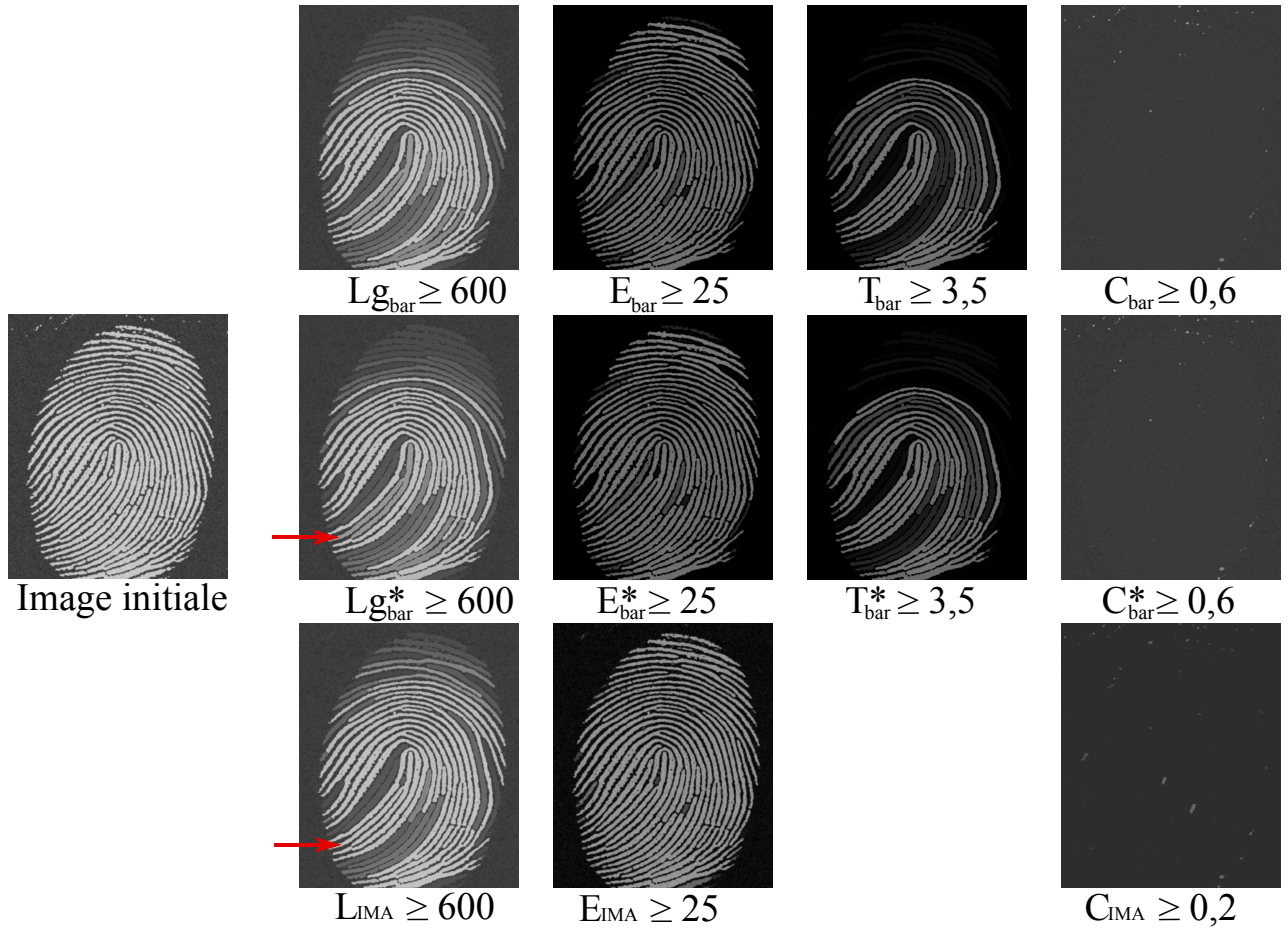


FIGURE 5.12 – Résultat des amincissements avec des attributs géodésiques basés sur la règle soustractive sur une image en niveaux de gris. De gauche à droite, l'image initiale, puis l'utilisation de critères de taille, d'allongement, de tortuosité et de circularité. Les lignes regroupent respectivement les attributs utilisant la distance géodésique, la distance géodésique généralisée et l'axe médian, filtré avec seuil adaptatif.

une même structure en comparaison avec  $L_{Bar}^*$ . L'apport de la mesure de longueur sur l'axe médian est montré pour la structure pointée par la flèche. Cette structure est l'union de deux fibres et la mesure  $L_{IMA}$  permet de le prendre en compte, préservant ainsi cette structure dans l'image. Néanmoins, le principal inconvénient de la méthode basée sur l'axe médian est son manque de précision pour mesurer les petites structures. En effet, on observe une sur-estimation de leur longueur limitant l'intérêt du filtre de circularité aux grandes structures.

### 5.7 Résultats

Pour visualiser les avantages des amincissements par attributs géodésiques, nous proposons tout d'abord une application dans le domaine bio-médical. La figure 5.13(a) montre une image d'ADN

acquise au microscope électronique<sup>1</sup>. Cette structure est longue et tortueuse et entre parfaitement dans le domaine de compétence des filtres présentés dans ce chapitre. La comparaison de la détection obtenue sera réalisée visuellement en comparant les amincissements par attributs géodésiques avec d'autres méthodes utilisées pour extraire le même type de structures :

1. **Le supremum des ouvertures avec des segments** orientés tout les 2 degrés (figure 5.13(b)). Comme on l'a vu dans le chapitre 2, seules les structures rectilignes de la molécule d'ADN sont extraites de l'image. La tortuosité est pour cette application, une donnée importante et ce filtre sous-estime très largement la longueur totale de cette structure. On note de plus que la taille du segment  $L$  est très faible en comparaison avec les autres méthodes. Une ouverture avec des segments de taille 100 pixels aurait complètement supprimé la molécule.
2. **Une ouverture par chemins** de taille 160 pixels (figure 5.13(c)). Une ouverture par chemins est un opérateur introduit dans le chapitre 3. Il permet d'extraire les structures dans l'image qui ne sont pas parfaitement rectilignes. Le graphe sous-jacent à l'image utilisé est celui présenté par Heijmans *et al.* dans [90] et repris dans ce document (3 prédécesseurs et 3 successeurs). La détection obtenue est bien meilleure que pour les ouvertures linéaires. Le bruit est considérablement réduit cependant, une partie de la molécule est supprimée. À cause de la tortuosité de la structure à détecter, cet opérateur ne peut pas la suivre complètement et sous-estime sa longueur réelle.
3. **Un amincissement avec le diamètre barycentrique** ( $Lg_{Bar}$ ) de taille 600 pixels. Ce filtre donne de très bon résultats (figure 5.13(d)). En effet, la molécule est extraite de l'image et plus généralement la longueur des structures très tortueuses est correctement estimée. Ainsi, ce filtre offre une détection plus flexible par rapport aux ouvertures par chemins puisque la contrainte sur la tortuosité est relâchée.
4. **Un amincissement avec l'élongation** ( $E_{Bar}$ ) de taille 50 (figure 5.13(e)). Cette méthode filtre le bruit et propose une très bonne détection. Toutes les structures de l'image qui ne sont pas allongées sont supprimées.
5. **Un amincissement avec la tortuosité** ( $T_{Bar}$ ) de taille 2,5 (figure 5.13(f)). Ce filtre supprime toutes les structures non tortueuses de l'image et préserve correctement la molécule.

Ces filtres sont maintenant utilisés pour la détection des fissures pour le contrôle non destructifs de pièces industrielles (chapitre 7). Les structures à détecter sont longues, fines et possèdent une tortuosité aléatoire. L'utilisation des outils développés dans ce chapitre est donc très adaptée. La figure 5.14 propose des résultats de détection des ces structures avec le même groupe d'opérateurs que pour l'extraction de la molécule d'ADN, en ajoutant trois autres attributs  $Lg_{Bar}^*$ ,  $L_{IMA}$  et  $C_{Bar}$ . Les mêmes conclusions peuvent être tirées : les filtres par attributs géodésiques proposent une bonne estimation de la longueur d'un objet même s'il est très tortueux. De plus, l'amincissement avec un attribut géométrique préserve d'avantage de structures dans l'image puisqu'il est capable de mesurer plus efficacement les structures qui forment un réseau.

---

1. Image provenant de l' *Institute for Molecular Virology, University of Wisconsin - Madison* <http://www.biochem.wisc.edu/faculty/inman/empics/dna-prot.htm>

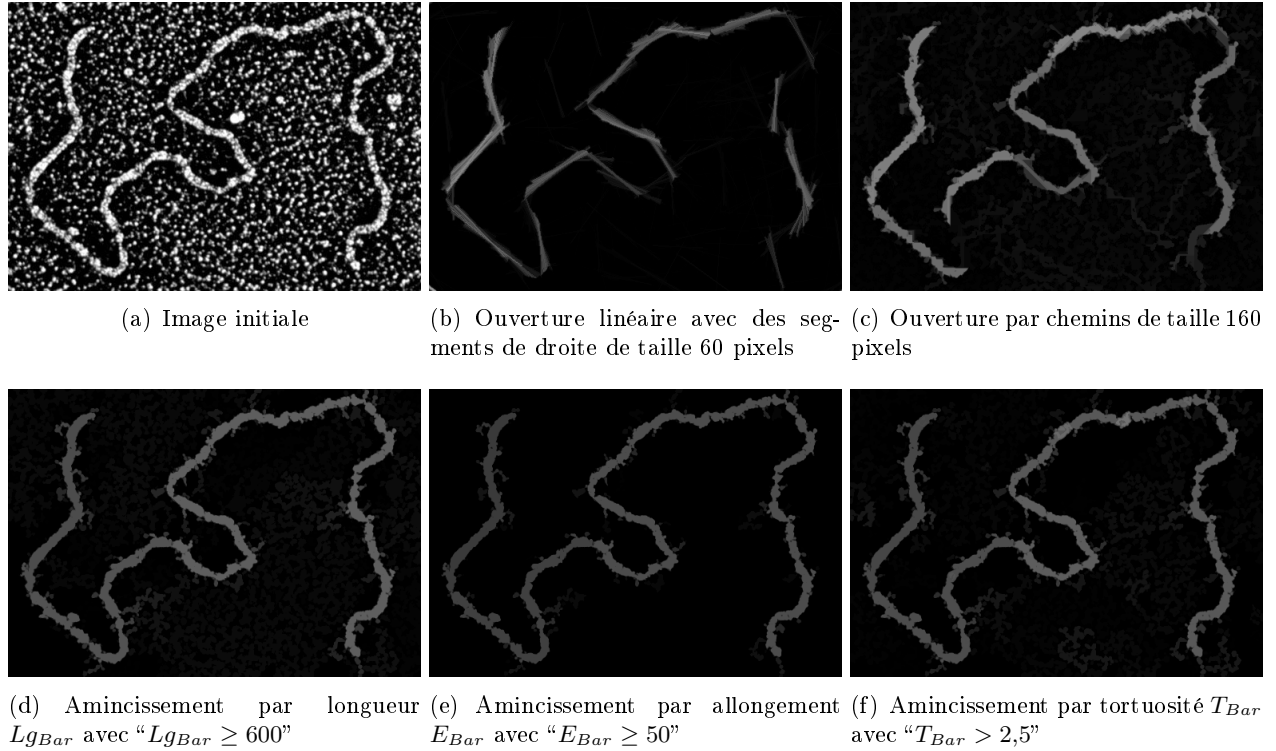


FIGURE 5.13 – Extraction d’une molécule d’ADN. Les résultats de détection de cinq méthodes sont comparés et on remarque que seuls les amincissements avec des attributs géodésiques ne sous-estiment pas la longueur de cette molécule.

### 5.8 Algorithme et considérations pratiques

Le temps de calcul est un élément central dans de nombreuses applications et particulièrement pour le contrôle non destructif avec un système de vision (cf. chapitre 7). Par conséquent, nous avons développé un nouvel attribut géodésique uniquement dans le but d’accélérer ces opérateurs (voir la section 5.4 et la description du diamètre barycentrique) et nous présentons dans cette section un algorithme rapide pour calculer les amincissements par attributs géodésiques. On comparera les temps d’exécution de cet algorithme avec ceux d’une implémentation naïve (utilisation du diamètre géodésique à partir de tous les points de contour) puis avec ceux des ouvertures par chemins.

#### 5.8.1 Algorithme des amincissements par attributs géodésiques

Les filtres connexes sont souvent implémentés en utilisant une structure d’arbre appelée max-tree ou arbre de composantes connexes (voir Salembier *et al.* [199]). La création du max-tree repose sur des inondations récursives en partant du point le plus bas de l’image. Pour une image,  $f : D \rightarrow V$ , la plus défavorable, la complexité de cet algorithme est en  $\mathcal{O}(N \times M)$ , avec  $M = \text{card}(V)$  et  $N = \text{card}(D)$ , voir [163]. Néanmoins, le pire des cas est rare en pratique et l’algorithme de Salembier *et al.* est plus rapide que celui de Najman et Couprie [163] pour des images 8 bits, pourtant conçu pour avoir une

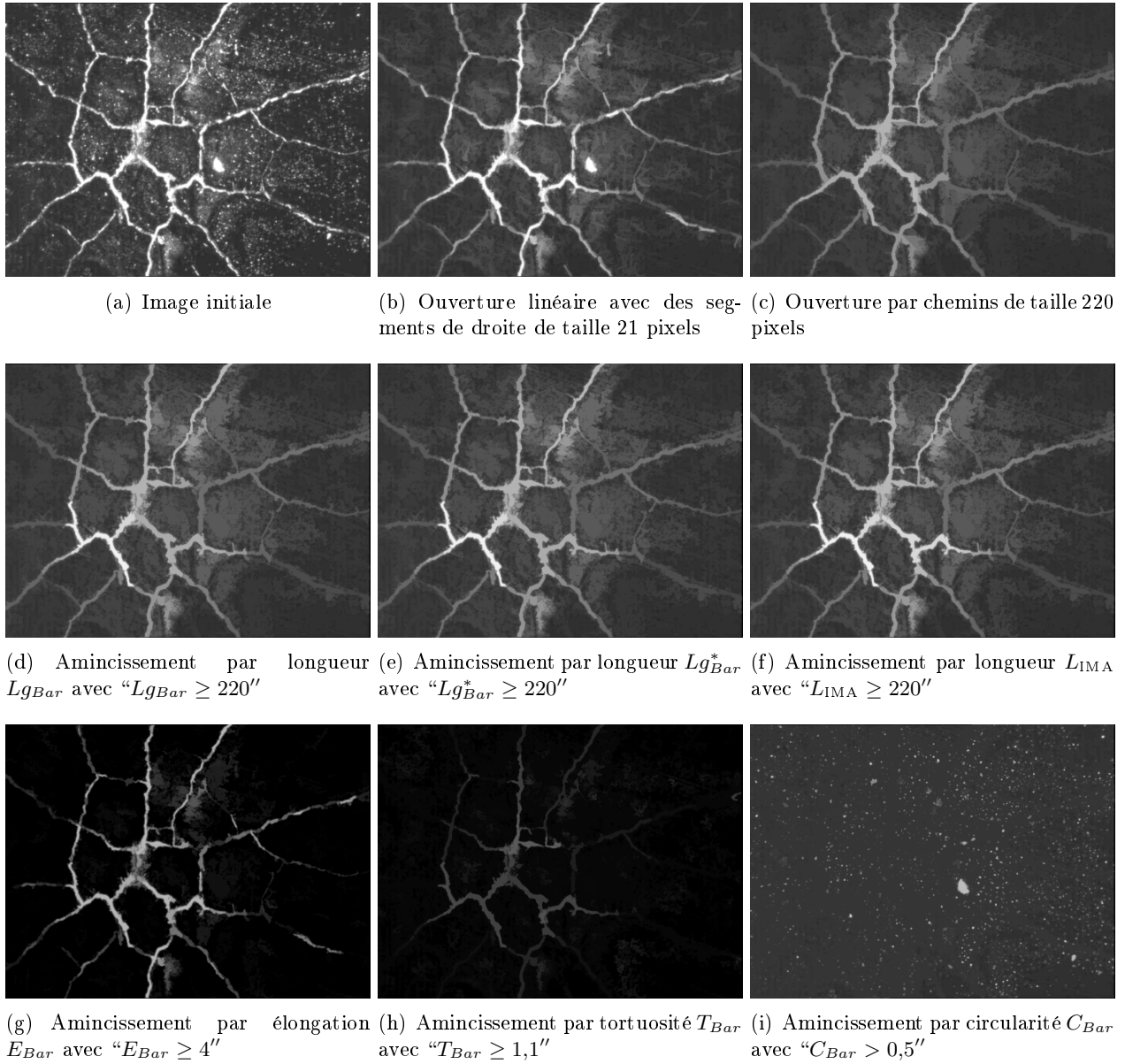


FIGURE 5.14 – Détection des fissures. 8 méthodes sont utilisées pour extraire les fissures de l'image. Les amincissements par attributs géodésiques fournissent la meilleure détection.

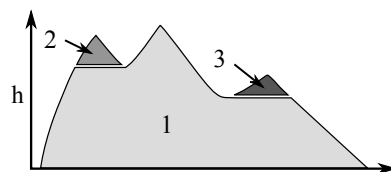


FIGURE 5.15 – Illustration sur un signal 1-D du principe d'extraction des composantes connexes. Les composantes en gris clair sont traitées en premier, puis celles en gris puis, dans un troisième temps, celles en gris foncé.

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSIQUES ET GÉOMÉTRIQUES

complexité inférieure, en temps quasi constant par pixel avec une approche *union find* ([228, 253, 92]). D'autres implémentations rapides existent, voir par exemple Ngan *et al.* [167] et Lesage *et al.* [120].

Ces structures de données avec des arbres ou des approches par *union find* sont adaptées lorsque les attributs peuvent être *mis à jour* à chaque fois qu'un nouveau pixel est ajouté à la composante connexe. Ce n'est pas le cas avec le diamètre barycentrique ou l'axe médian, qui doivent être *recalculés* et avoir accès à tous les pixels de la composante connexe. Aussi, cela interdit la récursivité ainsi que les approches de type *union find*, puisque l'accès à tous les pixels d'une même CC n'est pas facile.

À la place, on considère l'image 2-D comme un relief topographique où les hautes (resp. basses) valeurs de niveaux de gris correspondent aux montagnes (resp. vallées). On simule le processus d'inondation. On démarre avec le relief complètement submergé d'eau, et nous laissons progressivement le niveau de l'eau baisser, laissant apparaître le premier maximum (global). Sa composante connexe est progressivement reconstruite et testée avec  $\chi$ . Lorsque d'autres maxima régionaux apparaissent, ils ne sont pas traités tout de suite et nous reconstruisons toutes les composantes connexes à des niveaux de gris inférieurs (voir les composantes du bloc 1 de la figure 5.15). Enfin, les autres maxima locaux sont traités de la même façon mais l'agrégation s'arrête lorsqu'une composante connexe est déjà traitée (blocs 2 et 3 de la figure 5.15).

L'algorithme 10 simule ce processus. Il utilise une file à priorités *HQueue*, qui supporte les opérations suivantes : *Push(x,p)* (insertion du pixel  $x$  dans la file avec une priorité  $p$ ) et  $(x,p) = Pop()$  (suppression de l'élément de plus haute priorité de la file) ainsi que les fonctions accesseurs  $p = Prio()$  et *Empty()* qui retournent respectivement la priorité du prochain élément à dépiler et une variable binaire indiquant si *HQueue* est vide.

On commence par initialiser l'image de sortie avec l'image d'entrée (ligne 4) puis on extrait, avec la fonction *FindLocalMax* (ligne 5), un pixel par maximum régional, classé par ordre décroissant en fonction du niveau de gris.

Le cycle *For* des lignes 8 à 28 analyse successivement tous les maxima régionaux de l'image et la ligne 9 lit la position  $p$  du prochain maximum à traiter à partir de la liste *LocalMax*. Finalement, la boucle *Repeat-Until* (lignes 14 à 28) simule la descente du niveau de l'eau et teste chaque CC de l'image avec  $\chi$  (ligne 26).

L'extraction de la composante connexe nécessite deux tableaux temporaires, *State* :  $D \rightarrow \mathbb{Z}$  et *Level* :  $D \rightarrow B$  (lignes 2 et 3), initialisés à 0 pour tous les pixels de l'image (lignes 6 et 7). *State* reçoit  $-1$  à chaque fois qu'un pixel est inséré dans *HQueue* (ligne 11) puis, plus tard, il reçoit l'index du maximum local auquel il appartient (ligne 18), lorsqu'il est dépilé. On initialise le processus de croissance de région en insérant le pixel  $p$  dans *HQueue* avec une priorité égale à  $f(p)$  (ligne 10), en marquant ce pixel (lignes 11 et 12) et en initialisant *CC* à l'ensemble vide. Ainsi, le processus de croissance de région (ligne 15 à 25) permet l'extraction de la CC au niveau  $h$ , auquel  $p$  appartient. Le critère  $\chi$  a besoin d'avoir accès à tous les pixels de la composante connexe analysée et, lorsque le processus de croissance de région est fini, la variable *CC* contient l'ensemble des pixels tels que  $f(x) \geq h, \forall x \in CC$ . On teste alors cette composante connexe avec  $\chi$  et on applique la règle soustractive en supprimant le contraste de cette CC à tous ses pixels. La fonction *contrast(CC)* retourne la différence entre le niveau  $h$  et le plus haut niveau de gris des pixels voisins de *CC*.

Le même processus est répété jusqu'à ce que tous les maxima régionaux soient traités.

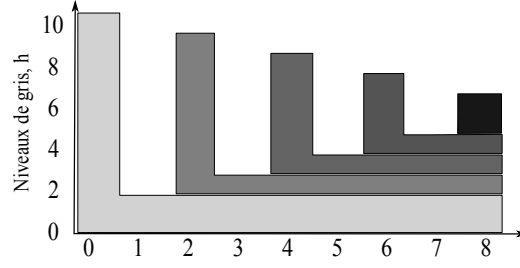


FIGURE 5.16 – Illustration de la décomposition du pire signal 1-D possible avec l’algorithme développé dans la section 5.8. On parcourt en moyenne  $N^2/4$  fois chaque pixel.

La fonction  $Lg_{Bar}$  calcule la valeur du diamètre barycentrique. Il identifie tout d’abord le barycentre de  $CC$ , écrit  $b$  (ligne 1), trouve le point le plus éloigné de  $b$  avec une distance euclidienne, ligne 2, puis il réalise deux propagations dans la composante connexe (voir l’équation 5.5 et les lignes 3 et 4) afin de retourner la valeur du diamètre barycentrique.

### 5.8.2 Complexité de l’algorithme

L’analyse de la complexité peut être décomposée en deux parties : (i) la complexité de l’extraction des composantes connexes (algorithme 10) et la complexité du calcul du diamètre barycentrique (algorithme 11).

Nous détaillons tout d’abord la complexité de l’algorithme 10. La fonction  $FindLocalMax$  remplit le tableau  $LocalMax$  en utilisant un processus de croissance de régions, associé à un algorithme de tri en temps constant (*radix sort* [170]). Cette fonction a donc une complexité linéaire par rapport au nombre de pixels de l’image,  $\mathcal{O}(N)$  (avec  $N = card(D)$ ). Enfin, à partir de tous les maxima locaux, on utilise un processus d’émersion du relief avec une file d’attente hiérarchique, qui ajoute un facteur  $\log(M)$  (avec  $M = card(V)$ ) dans la complexité totale de l’algorithme. Au pire des cas, pour le signal fortement imbriqué représenté dans la figure 5.16, chaque pixel est traité en moyenne  $N/4$  fois donnant une complexité en  $\mathcal{O}(N^2 \log(M))$ . Il s’agit de la même complexité que celle de l’algorithme de Breen et Jones [29], mais on note que les temps d’exécution observés sont loin de cette borne supérieure. En pratique, on observe une évolution linéaire des temps de calcul avec la taille de l’image et une expérience montre, sur une centaine d’images en niveaux de gris, que chaque pixel n’est traité que 2 fois en moyenne, très loin des  $N/4$  fois de la borne maximale.

Tous les attributs basés sur la distance géodésique ont besoins du calcul du diamètre barycentrique. Ainsi, la complexité de ces attributs est la même que celle de l’algorithme 11.

Si on considère une composante connexe ayant  $n$  pixels, le calcul du diamètre barycentrique est mené en 4 étapes : le calcul du barycentre nécessite un balayage de tous les pixels de la CC. La complexité est donc en  $\mathcal{O}(n)$ , tout comme la recherche du point le plus éloigné de ce barycentre. Enfin, deux propagations sont réalisées pour obtenir la valeur du diamètre barycentrique. Pour une propagation, suivant la distance choisie,  $\mathcal{L}_\infty$  ou  $\mathcal{L}_2$ , on utilise une file d’attente (ou une file d’attente prioritaire), donnant une complexité en  $\mathcal{O}(1)$  (ou en  $\mathcal{O}(\log(m))$ ) par pixel, avec  $m$ , la valeur moyenne des éléments dans la file.



## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSIQUES ET GÉOMÉTRIQUES

---

**Algorithme 10:**  $g \leftarrow \text{GEOATTRTHINNING}(f, \chi)$

---

```

Input :  $f : D \rightarrow V$  - image initiale
           $\chi$  - critère
Result :  $g : D \rightarrow V$  - image de sortie

  /* Déclaration des variables auxiliaires */
1  $HQueue = \{\}$  /* Initialiser  $HQueue$  à zéro */
2  $State : D \rightarrow \mathbb{Z}$ 
3  $Level : D \rightarrow V$ 

  /* Initialisation */
4  $g = f$ 
5  $LocalMax = FindLocalMax(f)$  /* Obtenir la liste triée des maxima locaux */
6  $State(\cdot) = 0$ 
7  $Level(\cdot) = 0$ 

  /* Pour tous les maxima locaux */
8 for  $i = 1 \dots Card(LocalMax)$  do
9    $p = LocalMax(i)$ 
10   $HQueue.Push(p, f(p))$ 
11   $State(p) = -1$  /* pixel empilé */
12   $Level(p) = f(p)$ 
13   $CC = \{\}$ 

  /* Simulation de la descente des eaux */
14 repeat
15   /* Extraction de CC qu'il faut tester avec  $\chi$  */
16   repeat
17      $(q, h) = HQueue.Pop()$ 
18      $CC = CC \cup \{q\}$ 
19      $State(q) = i$ 
20     foreach  $n \in Neighbour(q)$  do
21       if  $f(n) > Level(n)$  and  $State(n) \neq i$  then
22          $hmin = \min(h, f(n))$ 
23          $HQueue.Push(n, hmin)$ 
24          $State(n) = -1$  /* pixel empilé */
25          $Level(n) = hmin$ 
26   until  $h < HQueue.Prio()$ 
27   if not  $\chi(CC)$  then
28      $g(CC) = g(CC) - contrast(CC)$ 
29 until  $HQueue.Empty()$ 

```

---

---

**Algorithme 11:**  $l_x^2 \leftarrow Lg_{Bar}(CC)$ 


---

**Input :**  $CC$  - une composante connexe

**Result :**  $l_x^2$  : Diamètre barycentrique de  $CC$

```

1  $b = \bar{p}, p \in CC$ 
2  $x = \operatorname{argmax}_{y \in CC} ||y, b||$ 
3  $x^1 = \operatorname{argmax}_{y \in CC} d_{CC}(x, y)$ 
4  $x^2 = \operatorname{argmax}_{y \in CC} d_{CC}(x^1, y)$ 
5  $l_x^2 = d_{CC}(x^1, x^2)$ 
6 return  $l_x^2$ 
```

---

Ceci constitue une amélioration majeure en comparaison avec la complexité du calcul du diamètre géodésique. Avec une propagation à partir de tous les points du bord de l'objet, la complexité est  $\mathcal{O}(n^2)$  (l'objet le plus défavorable est une ligne qui nécessite  $n$  propagations). Avec la méthode de Schmitt, le nombre de propagations est réduit et est égal, au pire des cas, à  $\sqrt{n}$  pour un disque.

### 5.8.3 Optimisations

La première optimisation et probablement la plus importante en termes de gain de temps est l'arrêt de la propagation lorsque le critère est atteint. Cette optimisation est disponible uniquement pour les attributs basés sur la longueur et sur l'élongation.

L'accélération de cet algorithme provient aussi d'autres optimisations qui sont plus des astuces informatiques :

- Le barycentre et l'aire de la composante connexe sont des éléments qui sont calculés pendant le processus de croissance de régions afin de supprimer les balayages inutiles de la composante connexe dans l'algorithme 11.
- Dans cette même fonction, on n'utilise pas de carte de distance pour la phase de propagation. À la place, un pixel sera inséré dans une file avec sa valeur de distance.
- On définit des files d'attente spécialisées à notre algorithme. Ainsi, supprimer tous les éléments de la pile est une opération réalisée avec une unique assignation.
- Pour des images 8 bits, on fusionne les deux images temporaires *State* et *Level*. Le premier octet stocke la valeur de la variable *Level* et la variable *State* utilise les autres octets. Ainsi, nous obtenons la valeur des deux variables en un unique accès mémoire.

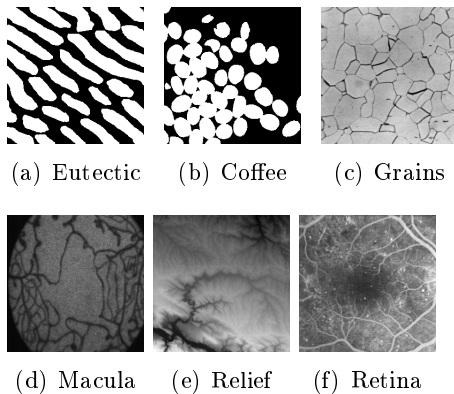
Utiliser l'algorithme 10 ainsi que le diamètre barycentrique et les optimisations présentés ci-dessus, accélèrent considérablement le calcul des amincissements par attributs géodésiques. La section suivante mesure ce gain.

### 5.8.4 Temps de calcul

On compare dans cette section, les temps de calcul des opérateurs introduits dans ce chapitre. Le calcul des amincissements utilisant le diamètre barycentrique avec une distance géodésique classique, sont plus simples si on les compare avec les attributs utilisant une distance géodésique généralisée ou l'axe médian. Ainsi, on propose tout d'abord, de mesurer cette différence pour 6 images arbitrairement choisies (images de la figures 5.17), enfin, on comparera les temps d'exécution avec les ouvertures par chemins.

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSQUES ET GÉOMÉTRIQUES

Le tableau 5.17 regroupe les temps d'exécution des amincissements utilisant 5 attributs différents : le diamètre géodésique  $Lg$  (calculé à partir de tous les points du contour de l'objet), le diamètre barycentrique  $Lg_{Bar}$ , le diamètre barycentrique avec une distance géodésique généralisée  $Lg_{Bar}^*$  et l'axe médian  $L_{IMA}$ .



Temps (ms)	$Lg$	$L_{Bar}$	$L_{Bar}^*$	$L_{IMA-Bar}$
Coffee	79	2,97	5,94	6,72
Eutectic	15	2,97	6,72	7,81
Grains	5400	42,34	334,07	515,78
Macula	2700	34,22	221,72	303,91
Relief	990	21,72	137,18	185,94
Retina	1560	24,69	134,53	197,34

FIGURE 5.17 – Temps de calcul des amincissements par attributs de longueur, pour différentes images de taille 256x256 pixels ( $L = 20$  pixels). Les temps sont en millisecondes, calculés avec un ordinateur portable Intel Core2 Duo T7700 @ 2.40GHz.

Comme attendu, les amincissements avec  $Lg_{Bar}$  présentent les meilleures temps de calcul et ceux-ci sont en moyenne, sur ces 6 images, 60 fois, 5 fois et 7 fois plus rapides que, respectivement,  $Lg$ ,  $Lg_{Bar}^*$  et  $L_{IMA}$ .

Les ouvertures par chemins sont des opérateurs développés pour répondre à des applications similaires à celles des amincissements par attributs géodésiques. La section 5.7 et le chapitre 3 montrent que la détection avec des ouvertures par chemins est bonne même si des structures plus tortueuses peuvent être détectées avec des filtres connexes. On compare ici leur temps d'exécution et la figure 5.18 mesure les temps moyens sur 100 itérations entre une implémentation rapide des ouvertures par chemins [224, 126] et des amincissements avec  $Lg_{Bar}$ . On utilise l'image des fissures de la figure 5.14 pour réaliser ce test (768x576 pixels) et on note que les amincissements par attributs géodésiques sont plus rapides que les ouvertures par chemins. On observe la même tendance pour d'autres images.

### 5.9 Bilan et perspectives

Ce chapitre présente trois nouveaux attributs : (i) l'indice de tortuosité géodésique, qui quantifie la tortuosité d'une structure, (ii) le diamètre barycentrique, qui mesure rapidement la longueur d'une structure. Ce dernier attribut est une approximation du diamètre géodésique, on donne donc l'erreur maximale théorique et on montre, sur plus de 50000 formes binaires, que l'erreur relative commise est négligeable (inférieure à 0,3%) et (iii) les attributs géométriques.

Ces attributs ainsi que d'autres tels que la longueur, l'élongation, la tortuosité ou la circularité, sont combinés aux amincissements par attributs afin de créer une nouvelle famille d'opérateurs. Ils permettent de facilement discriminer les structures longues, fines et/ou tortueuses et on montre la

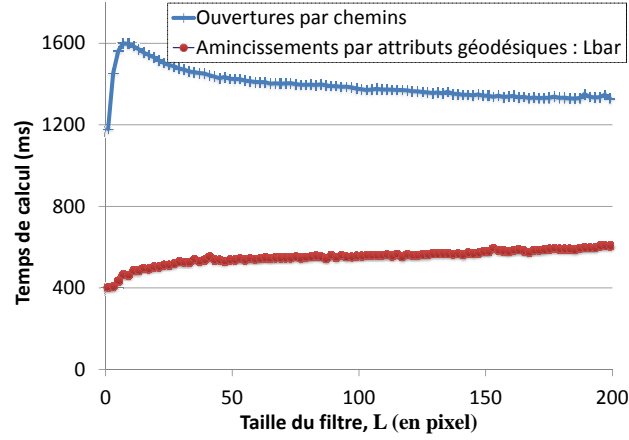


FIGURE 5.18 – Temps de calcul entre les ouvertures par chemins classiques et les amincissements par attributs géodésique  $Lg_{Bar}$  avec la règle de filtrage soustractive.

supériorité de ces opérateurs pour la détection des structures fines et très tortueuses, en les comparant aux ouvertures linéaires et aux ouvertures par chemins.

D'autres attributs basés sur la distance géodésique généralisée et sur l'axe médian ont été développés. Ils permettent d'améliorer la précision de ces opérateurs et de mesurer la longueur des réseaux de fibres. Ils sont cependant 5 à 7 fois plus lents que les amincissements avec des attributs géodésiques.

Nous utilisons la règle soustractive pour tous nos amincissements et nous donnons un algorithme efficace pour les calculer. On note que les temps de calcul sont compétitifs par rapport aux ouvertures par chemins et ces opérateurs peuvent donc être utilisés dans des applications industrielles où les contraintes temporelles sont importantes.

Pour le moment, les amincissements par attributs géodésiques sous-estiment la longueur des structures déconnectées par du bruit. On pourra résoudre ce problème en utilisant une connectivité de seconde génération, méthode définie par Braga-Neto et Goutsias [27] ou par Ouzounis et Wilkinson dans [174]. Enfin, pour étendre ce travail, on peut généraliser ces outils pour créer des ouvertures ultimes [19, 91] ou des courbes granulométriques en une seule immersion de l'image.

## 5. AMINCISSEMENTS PAR ATTRIBUTS GÉODÉSIQUES ET GÉOMÉTRIQUES

---

---

Analyse statistique parcimonieuse : AdaCOS, nouvelle méthode de classification qui minimise une fonction de coût lors de la génération d'un modèle.

---

« *La simplicité est la sophistication suprême* »  
Léonard de Vinci

En statistique, la fouille ou l'exploration des données<sup>1</sup> permet d'extraire *automatiquement* une connaissance à partir d'une grande quantité de données. Ces méthodes sont divisées en deux groupes :

1. **Les méthodes descriptives** sont aussi appelées méthodes de *classification*<sup>2</sup>. Elles permettent d'organiser, de simplifier et de regrouper des données hétérogènes en sous groupes homogènes. En voici quelques exemples : analyse en composantes principales, analyse factorielle, moyennes mobiles, regroupement hiérarchique... (voir le projet Colas, chapitre 9). Ces méthodes descriptives réalisent un apprentissage dit *non supervisé*, puisqu'elles ne nécessitent aucune information *a priori* de la part de l'utilisateur.
2. **Les méthodes prédictives** sont des méthodes de *classement*<sup>3</sup>. Elles réalisent un *apprentissage dit supervisé*, puisqu'elles établissent des règles et des modèles à partir d'exemples déjà traités et validés par des experts (régression linéaire, régression logistique, machine à noyaux (SVM), forêt aléatoire...)

En classement, la régression linéaire est la plus simple et la plus répandue des méthodes. Elle minimise une fonction *erreur* afin d'avoir un estimateur non biaisé. Il est cependant bien connu que l'on peut améliorer le pouvoir de prédiction de cette méthode en pénalisant les coefficients de l'estimation.

Dans ce document, on introduit un modèle de classement linéaire avec une nouvelle pénalisation, permettant de répondre à un nouveau type de problème : comment créer un modèle prédictif précis, qui minimise en même temps une fonction de coût indépendante des données. Cette fonction de coût sera par exemple le temps nécessaire pour calculer tel ou tel descripteur, afin de choisir celui qui sera le plus rapide, à performance égale (projet TOCATA, chapitre 7). AdaCOS (**A**daptive **C**oefficient **S**hrinkage) adapte la pénalisation pour chaque descripteur et correspond à une généralisation de la régression Ridge.

---

1. Exploration des données : *data mining* en Anglais  
2. Classification : *clustering* en Anglais  
3. Classement : *classification* en Anglais

Enfin, nous étendons la procédure *forward stagewise* (méthode utilisée pour résoudre les régressions pénalisées) pour obtenir un modèle parcimonieux (où seuls quelques descripteurs seront utilisés dans la régression). On obtient le *Forward AdaCOS* qui pourra être utilisé dans des problèmes de minimisation de temps, de coût ou d'autres problèmes d'optimisation. De plus, la méthode AdaCOS peut être utilisée astucieusement avec les méthodes de régression parcimonieuses LARS et GLMNET afin d'obtenir une implémentation rapide et généralisable.

**Mots clés :** Analyse statistique, AdaCOS, pénalisation, pénalisation adaptative, régression ridge, forward selection, optimisation, sélection de variables, LARS, GLMNET.

### 6.1 Introduction et motivations

En analyse statistique, les modèles linéaires ont été largement étudiés depuis plus de 50 ans et sont utilisés avec succès dans de nombreuses applications dans des domaines très variés. On décrit ici la régression linéaire, puis l'apport des pénalisations sur la régression.

#### 6.1.1 Notations

On utilise les notations classiques des méthodes d'analyse statistique [87]. Soit  $N$  et  $p$ , respectivement le nombre d'instances et le nombre de descripteurs dans la base de données. Une variable d'entrée sera notée par le symbole  $X$ , une variable de sortie quantitative par  $Y$  et catégorielle par  $G$ . Si  $X$  est un vecteur, alors on accède au  $j^{\text{ième}}$  élément de  $X$  par l'indice  $x_j$ . On utilise les majuscules lorsque l'on parle des aspects généraux d'une variable puis les minuscules pour les observations. Une matrice sera toujours notée en majuscule et en gras, et les vecteurs de taille  $N$  seront aussi en gras pour faire la distinction entre un vecteur  $\mathbf{x}_i$  et une observation  $x_i$ .

Enfin,  $\hat{Y}$  prend des valeurs dans  $\mathbb{R}$  et correspond aux prédictions liées aux données  $X$ . De même,  $\hat{G}$  prend ses valeurs dans le même ensemble que  $G$  et lorsque celui-ci n'aura que deux classes, on le traitera comme une variable quantitative. En effet, les prédictions prendront des valeurs comprises entre 0 et 1 et on retrouve la catégorie  $\hat{G}$  en posant  $\hat{Y} \geq 0,5$ .

#### 6.1.2 Régression linéaire

Soit  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$ , une matrice de taille  $N$  par  $p$ , avec  $\mathbf{x}_j, \forall j \in (1, \dots, p)$  un vecteur de taille  $N$ . Dans la suite de ce document, on normalise<sup>1</sup> cette matrice afin de pouvoir comparer plus facilement les variations des descripteurs entre eux, même s'ils n'ont pas les mêmes unités. Un modèle linéaire est alors défini de la manière suivante :

$$\hat{\mathbf{y}} = \mathbf{X}\hat{B}, \quad (6.1)$$

avec  $\hat{B}$ , un vecteur de taille  $p$  correspondant aux coefficients du modèle. La méthode la plus simple pour caler un modèle utilise la méthode des moindres carrés (*Ordinary Least Square* OLS). Cette méthode revient à trouver les coefficients  $B$  qui minimisent l'erreur quadratique du modèle entre les observations réelles et les observations prédites, soit :

$$\hat{B}^{OLS} = \underset{B}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (6.2)$$

---

1. Une matrice est normalisée ou standardisée si elle est centrée et réduite. La valeur moyenne et l'écart type de chaque colonne sont égaux respectivement à 0 et 1.

En notation matricielle, l'équation 6.2 s'écrit de la manière suivante :

$$\hat{B}^{OLS} = \underset{B}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}B)^T (\mathbf{y} - \mathbf{X}B), \quad (6.3)$$

avec  $(.)^T$  la matrice transposée. Cette équation est une fonction quadratique en  $p$  paramètres que nous pouvons dériver par rapport à  $B$  pour en trouver le minimum :

$$\frac{d((\mathbf{y} - \mathbf{X}B)^T (\mathbf{y} - \mathbf{X}B))}{dB} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}B). \quad (6.4)$$

Les paramètres qui minimisent l'erreur quadratique,  $B^{OLS}$ , seront déterminés en annulant la dérivée première, donnant analytiquement une unique solution :

$$Eq. 6.4 = 0 \quad \Leftrightarrow \quad \hat{B}^{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}, \quad (6.5)$$

si et seulement si la matrice  $\mathbf{X}^T \mathbf{X}$  est inversible. Cela signifie que le rang de  $\mathbf{X}^T \mathbf{X}$  doit être maximal (que les colonnes soient toutes linéairement indépendantes).

### 6.1.3 Régression linéaire pénalisée

La régression linéaire OLS construit un modèle non biaisé. Cependant, il est connu dans la littérature que l'on peut augmenter le taux de prédiction si on accepte d'introduire un biais dans l'estimateur.

#### Régression Ridge

Hoerl et Kennard [95] ont introduit une pénalisation pour la régression, appelée *Ridge*. La régression Ridge minimise la somme des erreurs quadratiques, comme pour la régression linéaire, tout en s'assurant que la somme des  $(B_j)^2$  soit inférieure à un paramètre donné :  $\sum_i (B_j)^2 \leq s$  avec  $s \in \mathbb{R}^+$ . Ce paramètre  $s$  contrôle la pénalisation du système. Plus  $s$  est faible, plus la pénalisation est forte. Au contraire,  $s \geq \sum_j (B_j^{OLS})^2$  signifie qu'aucune pénalisation n'est insérée sur les  $B_j$  et on obtient les mêmes résultats qu'avec la régression linéaire non pénalisée. Ce problème est formulé de la manière suivante rendant explicite la contrainte de taille :

$$\begin{aligned} \hat{B}^{Ridge} = \underset{B}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ \text{avec } \sum_{j=1}^p (B_j)^2 \leq s, \end{aligned} \quad (6.6)$$

ou avec une notation matricielle équivalente :

$$\hat{B}^{Ridge} = \underset{B}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}B)^T (\mathbf{y} - \mathbf{X}B) + \lambda B^T B, \quad (6.7)$$

avec  $\lambda \geq 0$ , une variable ayant un rôle symétrique à  $s$ . Cette pénalisation  $L_2$  applique un rétrécissement continu des valeurs des paramètres  $B_j$  vers zéro. Quand de nombreuses variables sont très corrélées, les coefficients du modèle peuvent devenir instables avec de fortes variances. Une très grande valeur d'un coefficient peut voir son effet annulé par une très forte valeur négative d'une variable corrélée. En imposant une contrainte de taille, ce phénomène ne se produira plus, contribuant à l'amélioration des



performances de prédiction du modèle à travers un compromis entre la variance et le biais (voir [87] chapitre 7). L'équation 6.7 est dérivable et possède la solution analytique suivante :

$$\hat{B}^{Ridge} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T y. \quad (6.8)$$

La matrice  $\mathbf{I}$  de taille  $p \times p$  correspond à la matrice identité. La méthode Ridge ajoute alors  $\lambda$  sur la diagonale de la matrice  $\mathbf{X}^T \mathbf{X}$  avant de l'inverser. Ceci améliore la stabilité de la régression linéaire, puisque l'inversion de cette matrice est toujours possible si  $\lambda > 0$ . Ce fut la motivation principale de Hoerl et Kennard.

Cet estimateur est facile à calculer puisqu'il possède une solution analytique, améliore le taux de prédiction et utilise généralement tous les descripteurs de la matrice  $\mathbf{X}$

### DÉFINITION 6.1

Le chemin de régularisation est l'ensemble des valeurs des coefficients  $B_j$  obtenues en faisant varier le paramètre de la régression :  $\lambda$ . Pour des régressions linéaires, le chemin de régularisation de chaque variable est linéaire par morceau [193].

Pour la régression Ridge, les chemins de régularisation des coefficients (figure 6.1(a)) est tracé en fonction de  $\sum_j |B_j| / \sum_j |B_j^{OLS}|$  pour normaliser le graphique entre 0 et 1.

### Régression Lasso

Outre l'amélioration du taux de prédiction, d'autres critères peuvent être utilisés pour évaluer la qualité d'un modèle. La simplicité et l'interprétabilité sont des notions clés pour certains chercheurs. La parcimonie met en relief d'une manière claire les liens existant entre les réponses et les variables, spécialement lorsque le nombre de descripteurs est important. Voir [83] et [94] pour un état de l'art des différentes techniques permettant de faire de la sélection de variables.

La régression *Lasso*<sup>1</sup>, introduite par Tibshirani en 1996 [233], est une méthode très similaire à la régression Ridge avec une différence subtile mais fondamentale, la norme  $L_2$  a été remplacée par la norme  $L_1$  :

$$\begin{aligned} \hat{B}^{Lasso} = \underset{B}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ \text{avec } \sum_{j=1}^p |B_j| \leq s. \end{aligned} \quad (6.9)$$

Cette contrainte rend les solutions non linéaires avec les  $y_i$  et des algorithmes d'optimisation convexe doivent être utilisés pour calculer les  $B_j$ . Cette norme  $L_1$  contraint certains coefficients à être exactement égaux à 0 et réalise une sélection de variables *continue*. Lorsque  $s \geq \sum_j |B_j|$ , alors aucune contrainte n'est appliquée sur les paramètres et on obtient  $B^{Lasso} = B^{OLS}$ . Le chemin de régularisation de la régression LASSO est présenté dans la figure 6.1(b).

### Régression Forward stepwise, forward stagewise

D'autres méthodes peuvent être utilisées pour obtenir un modèle parcimonieux. *Forward stepwise* est une méthode gloutonne qui incorpore successivement la variable la plus corrélée avec les résidus et

1. La régression LASSO est aussi appelée *basis pursuit* par Chen *et al* [36], dans la communauté du traitement du signal

l'ajoute dans le modèle (voir [87], section 3.2.2). La méthode *Forward Stagewise* propose une procédure moins brutale (voir [87] section 3.3.3) : au lieu d'ajouter à chaque itération la solution OLS de la variable la plus corrélée, on lui ajoute uniquement une petite valeur. Ainsi, une variable peut être sélectionnée un nombre important de fois tant qu'elle a une corrélation plus importante que les autres descripteurs. Cette procédure ne s'arrête que lorsque les corrélations de toutes les variables sont nulles (figure 6.1(d)). Cette méthode gloutonne est particulièrement adaptée et robuste au sur-apprentissage [83, 187].

## Régression LAR

Avec le même objectif, LAR (Least Angle Regression), introduit par Efron *et al.* [57], permet de tracer entièrement le chemin de régularisation d'une manière analytique. La complexité algorithmique de cette méthode est équivalente à celle de la régression linéaire classique et de plus, Efron *et al.* firent le lien direct entre les méthodes LAR, LASSO et Forward stagewise, qui produisent des chemins de régularisation très similaires (voir la figure 6.1). Aucune de ces trois méthodes ne semble avoir un clair avantage sur les autres, aussi, ils élaborèrent un algorithme générique permettant de les calculer efficacement d'une manière analytique.

## Autres pénalisations

D'autres approches peuvent être utilisées introduisant des comportements différents. Zou et Hastie proposent l'*Elastic Net* [258] afin de dicter un effet intermédiaire entre la régression Ridge et la régression Lasso. Frank et Friedman ont introduit la régression *Bridge* [68], Meinshausen le *relaxed LASSO* [140], Zou l'*adaptive LASSO* [257] et beaucoup d'autres (*group LASSO* [255], *bolasso* [8], *SCAD*, [59], l'*adaptive elastic net* [75], l'*adaptive Ridge* [6], FLASH [185]).

La majorité des pénalisations rencontrées dans la littérature ont été introduites pour améliorer le pouvoir de prédiction, la convergence ou la parcimonie des régressions linéaires. Dans ce chapitre, la pénalisation introduite est tout autre et dépend d'éléments extérieurs (une fonction coût) que nous cherchons à minimiser. Il est important de noter que la minimisation de cette fonction se fera au détriment de la précision du modèle. Il s'agira donc de trouver un compromis entre la précision du modèle et la minimisation de cette fonction coût.

### 6.1.4 Fonction de coût avec les régressions linéaires pénalisées

Nous avons vu que pour évaluer la qualité d'un modèle, on peut travailler sur la précision des prédictions et/ou la parcimonie du modèle. Pour notre application, nous introduisons un troisième critère, qui est en quelque sorte une extension du deuxième : le modèle généré sera parcimonieux mais il minimisera une fonction de *coût*, qui dépend d'une information donnée *a priori* sur chaque descripteur.

Ainsi, chaque variable va être pénalisée par une valeur qui peut être le coût en temps, le coût en argent, le coût en difficulté, ..., afin d'obtenir ce descripteur. Lorsqu'un descripteur est utilisé dans un modèle, cette valeur sera ajoutée à une coût global que l'on cherche à minimiser. Ainsi, nous cherchons à réaliser un compromis entre la précision du modèle et la minimisation de cette fonction.

Pour le projet TOCATA, décrit dans le chapitre 7, nous disposons d'un ensemble de descripteurs visant à extraire les défauts de surface d'une pièce métallique. Un modèle statistique sera utilisé pour obtenir un modèle précis. Cependant, de fortes contraintes de temps nous obligent à faire de la sélection de variables : sélectionner les descripteurs les plus pertinents, et si possible, les plus rapides à être

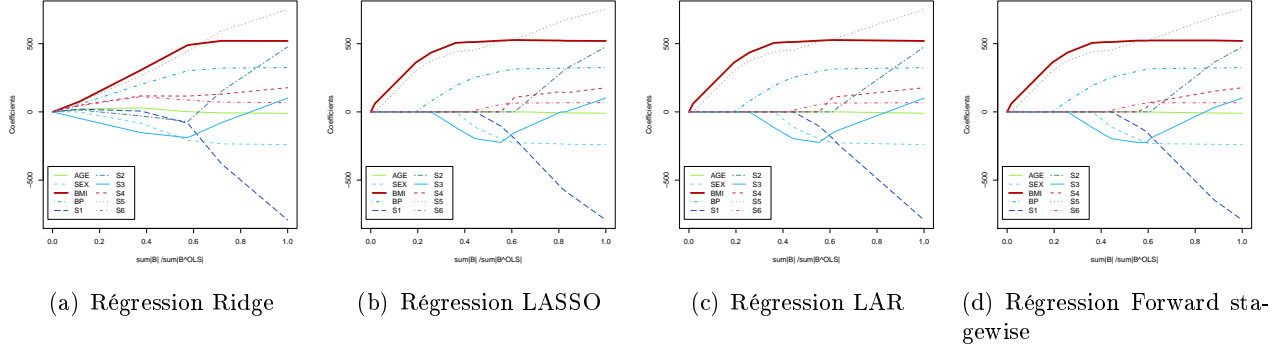


FIGURE 6.1 – Chemin de régularisation : régression Ridge, LASSO, LAR et Forward stagewise sur la base de données *Diabète*. Seules les méthodes LASSO, LAR et Forward stagewise proposent une solution parcimonieuse.

calculés. Nous utiliserons donc la méthode développée dans ce chapitre, couplée à une fonction de coût qui correspondra au temps nécessaire pour calculer les descripteurs.

Pour illustrer l'intérêt de cette méthode, nous choisissons une base de données standard en analyse statistique : la base de données “*Diabète*” [57], puis nous décrirons l'utilisation de cette méthode dans un contexte industriel avec le projet TOCATA.

La base de données *Diabète* est composée de 10 descripteurs : l'âge, le sexe, l'indice de masse corporelle (BMI), la pression sanguine (BP) et six sérums sanguins (S1 à S6). Tous ces descripteurs sont utilisés pour prédire l'évolution de la maladie une année après une première mesure. Cette étude a été menée auprès de 442 patients américains souffrant de diabète. Les chemins de régularisation de différentes méthodes sont présentés dans la figure 6.1.

## 6.2 Nouvelle méthode de classification : AdaCOS

Dans la formulation de la régression Ridge, le coefficient de chaque descripteur est pénalisé par le même paramètre  $\lambda$ . L'idée de la méthode AdaCOS est de pénaliser différemment chaque descripteur avec un coût que l'on note  $\alpha_1, \dots, \alpha_p$ . On obtient donc une nouvelle formulation de la régression linéaire pénalisée :

$$\begin{aligned} \hat{B}^{AdaCOS} &= \underset{B}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &\text{avec } \sum_{j=1}^p w_j (B_j)^2 \leq s, \end{aligned} \quad (6.10)$$

avec  $w_j$  les poids normalisés, puisque nous sommes uniquement intéressé par la différence relative entre deux descripteurs :

$$w_j = \left( 1 + \frac{\alpha_j}{\sum_{k=0}^p \alpha_k} \right)^\gamma, \quad (6.11)$$

avec  $\gamma$  un paramètre permettant d'accroître ou de réduire l'impact des coûts  $\alpha_j$  sur la régression. On écrit  $\mathbf{W}$  la matrice diagonale de taille  $p \times p$  où  $W_{jj}$  est égale à  $w_j$ . Ainsi, l'équation 6.10 s'écrit en notation matricielle :

$$\hat{B}^{AdaCOS} = \underset{B}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}B)^T (\mathbf{y} - \mathbf{X}B) + \lambda B^T \mathbf{W} B, \quad (6.12)$$

avec  $\lambda$  le paramètre de pénalisation. De la même manière que la méthode Ridge, l'équation 6.12 est linéaire et peut donc être dérivée :

$$\frac{d(y - \mathbf{X}B)^T (y - \mathbf{X}B) + \lambda B^T \mathbf{W} B}{dB} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}B) + \lambda (\mathbf{W} + \mathbf{W}^T) B. \quad (6.13)$$

$\mathbf{W}$  est une matrice diagonale donc  $\mathbf{W} + \mathbf{W}^T = 2\mathbf{W}$  et la solution analytique de l'équation 6.12 est donnée en annulant la dérivée première :

$$Eq. 6.13 = 0 \quad \Leftrightarrow \quad \hat{B}^{AdaCOS} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{W})^{-1} \mathbf{X}^T \mathbf{y}. \quad (6.14)$$

On remarque que la solution analytique de la méthode AdaCOS est très proche de celle de la méthode Ridge. Il s'agit donc d'une généralisation de la méthode OLS ainsi que de la méthode Ridge. En effet, si  $\mathbf{W} = 0$  on obtient  $\hat{B}^{AdaCOS} = \hat{B}^{OLS}$  et si  $\mathbf{W} = \mathbf{I}$  alors  $\hat{B}^{AdaCOS} = \hat{B}^{Ridge}$ .

Le nouveau paramètre introduit dans l'équation 6.11,  $\gamma$ , contrôle la différence relative entre les différents coûts de chaque descripteur. Lorsque  $\gamma > 1$ , les différences sont accrues et pour  $\gamma < 1$ , les différences sont réduites. Enfin, lorsque  $\gamma = 0$ , la fonction de coût n'a plus d'impact et on a exactement la même formulation que la méthode Ridge. Ce paramètre contrôle, d'une manière continue, l'importance de la fonction de coût dans la régression.

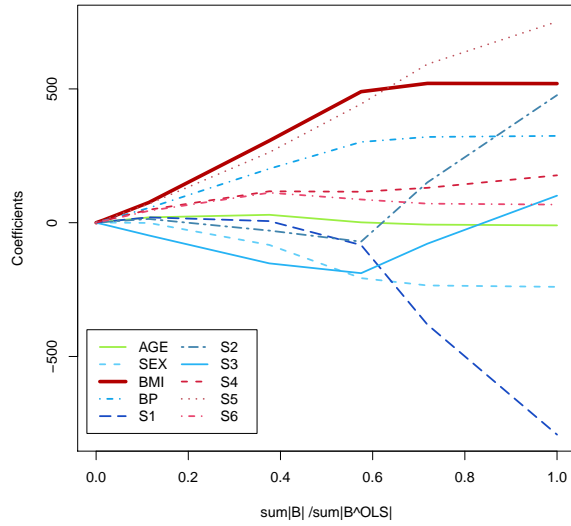
La figure 6.2 présente quatre chemins de régularisation de la méthode AdaCOS pour quatre valeurs différentes de  $\gamma$ . Pour visualiser l'impact de cette fonction de coût dans la régression, seule la variable ayant la plus haute corrélation avec les  $\mathbf{y}$  est pénalisée : BMI. Ce cas est hypothétique mais instructif puisque l'on remarque que l'influence de cette variable diminue lorsque  $\gamma$  augmente. Néanmoins, de la même manière que la régression Ridge, cette méthode n'est pas parcimonieuse et ne permet pas de réduire la fonction de coût globale, tous les descripteurs étant utilisés dans le modèle final. Aussi, on propose dans la section suivante de produire des modèles parcimonieux en utilisant un modèle AdaCOS.

### 6.3 Nouvelle méthode de classification parcimonieuse : Forward AdaCOS

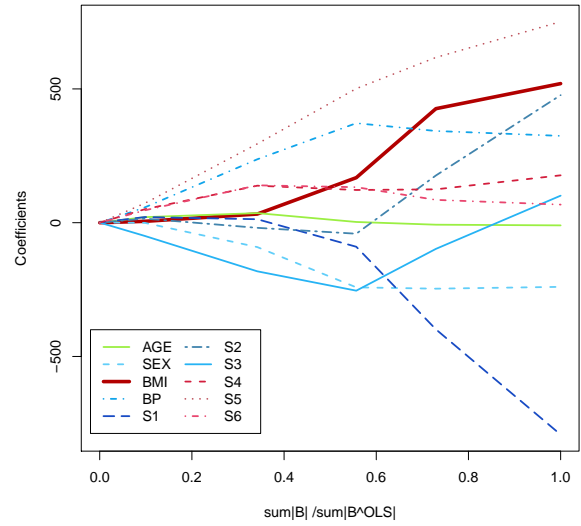
Dans la littérature, une des méthodes communément utilisées pour créer un modèle parcimonieux et faire de la sélection de variables consiste à utiliser des méthodes gloutonnes.

La méthode *Forward Stepwise* débute avec un modèle vide. À partir de tous les descripteurs, celui qui aura la plus grande corrélation avec la sortie  $\mathbf{y}$  en valeur absolue, sera sélectionné (on note  $\mathbf{x}_j$  ce descripteur). On calcule alors une régression linéaire produisant un vecteur résiduel, orthogonal à  $\mathbf{x}_j$ , qui est considéré comme notre nouvelle réponse. Ce processus est répété jusqu'à ce que tous les descripteurs soient insérés dans le modèle. La méthode forward stepwise réalise le meilleur mouvement à chaque itération et il est qualifié, pour cela, d'ajustement agressif.

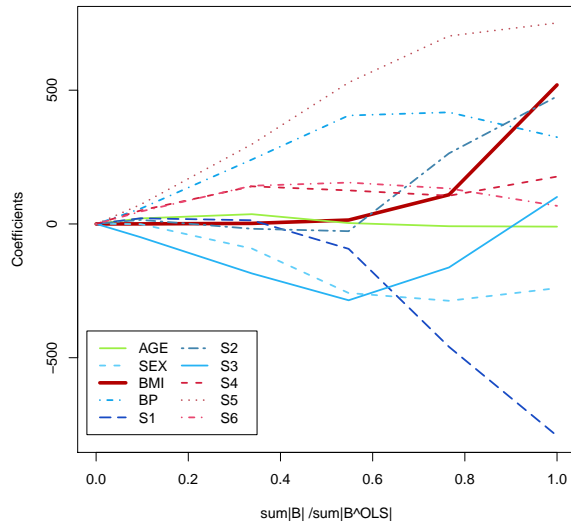
## 6. ADACOS



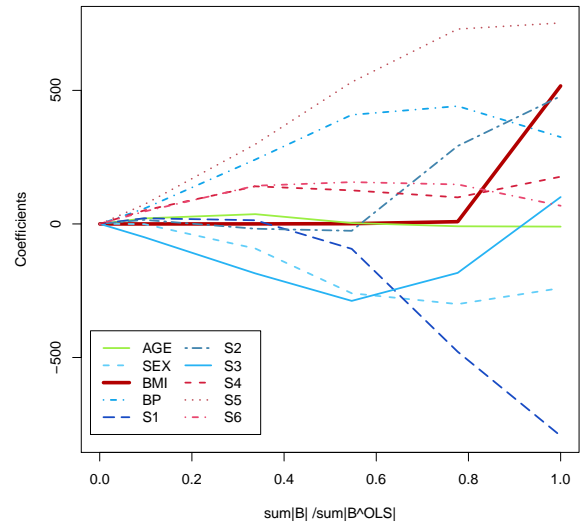
(a) Chemin de régularisation de la régression AdaCOS avec  $\gamma = 0$



(b) Chemin de régularisation de la régression AdaCOS avec  $\gamma = 2$



(c) Chemin de régularisation de la régression AdaCOS avec  $\gamma = 4$



(d) Chemin de régularisation de la régression AdaCOS avec  $\gamma = 6$

FIGURE 6.2 – Chemins de régularisation de la méthode AdaCOS sur l'étude *Diabète* avec 4 valeurs de  $\gamma$  différentes :  $\gamma = \{0, 2, 4, 6\}$ . Pour visualiser le fonctionnement de cette méthode, seule la variable BMI est pénalisée :  $\alpha = \{0, 0, 1, 0, 0, 0, 0, 0, 0, 0\}$ .

La régression *forward stagewise* agit avec plus de précautions. Au lieu de réaliser le meilleur mouvement à chaque itération, on ajoute au descripteur sélectionné une petite valeur. Une même variable peut donc être sélectionnée plusieurs fois consécutivement et cette procédure nécessite des centaines d'itérations avant de pouvoir construire le chemin de régularisation complet. Cette méthode est une approximation de la régression LASSO [87] et surpasse la régression Forward Stepwise.

Ainsi, nous considérons une version modifiée de la régression Forward Stagewise que nous appelons *Forward AdaCOS*. L'algorithme 12 résume le processus entier : le modèle est vide (ligne 1), et nous ajustons un modèle AdaCOS à chaque descripteur individuellement (lignes 4 et 5). Nous retenons le descripteur qui a la plus grande corrélation avec  $r$  en prenant en compte la fonction de coût de chaque descripteur. Les lignes 2 à 7 sont répétées tant que les corrélations avec  $r$  ne sont pas nulles. En pratique, les corrélations ne seront jamais strictement nulles, le pas *step* ayant une valeur fixe. Le processus s'arrête donc lorsque les coefficients de la régression sont égaux, à *step* près, à ceux de la régression linéaire non pénalisée. On note que si  $\gamma$  est égal à 0, alors le chemin de régularisation construit est identique à celui de la régression Forward Stagewise. Par conséquent, Forward AdaCOS généralise la méthode Forward Stagewise.

La figure 6.3 propose quatre chemins de régularisation pour des modèles Forward AdaCOS pour quatre valeurs différentes de  $\gamma$ . Pour  $\gamma = 0$ , on montre que l'on obtient le même résultat que la méthode Forward stagewise. Cependant, lorsque  $\gamma$  augmente, l'entrée dans le modèle de la variable BMI (la seule variable pénalisée) est retardée. Même si ce descripteur est le plus corrélé avec  $y$ , la fonction de coût réduit son impact d'une manière continue, grâce au paramètre  $\gamma$ .

---

**Algorithme 12:**  $\hat{B}^{FAdaCOS} \leftarrow \text{ForwardAdaCOS}(X, y, \text{step}, \alpha)$

---

**Input :**  $\mathbf{X}$  : matrice  $N \times p$ , descripteurs  
 $\mathbf{y}$  : vecteur de taille  $N$ , vérité terrain  
 $\alpha$  : vecteur de taille  $p$ , pénalisation de chaque descripteur  
**step** : pas d'incrément  
**Result :**  $\hat{B}^{FAdaCOS}$

---

```

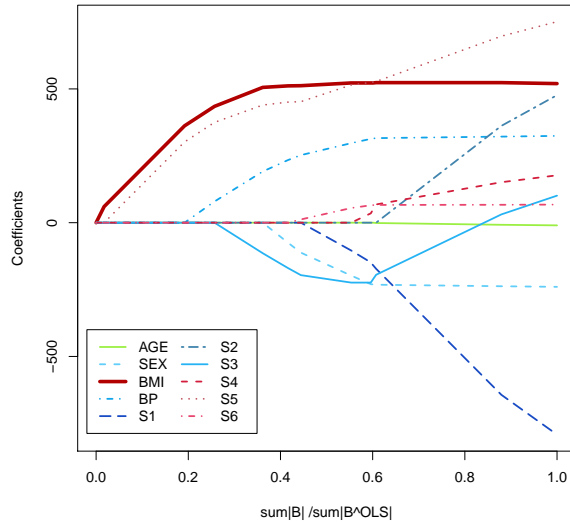
/* Initialisation                                                    */
1  $\hat{B}^{FAdaCOS} \leftarrow 0$ 
/* Construction du chemin de régularisation                          */
2 while  $B^{FAdaCOS} \neq B^{OLS}[\text{step}]$  do
3    $r \leftarrow (\mathbf{y} - \mathbf{X}\hat{B}^{FAdaCOS})$ 
4   for  $j = 1$  to  $p$  do
5      $B_j \leftarrow \text{AdaCOS}(r \sim x_j, \alpha)$ 
6     /*  $\sim$  syntaxe du langage R (http://www.r-project.org/) qui signifie que
7       l'on explique  $r$  par la variable explicative  $x_j$  */
6    $j = \text{argmax}_{j \in 1, p} |B_j|$ 
7    $\hat{B}_j^{FAdaCOS} \leftarrow \hat{B}_j^{FAdaCOS} + \text{sign}(B_j) \times \text{step}$ 

```

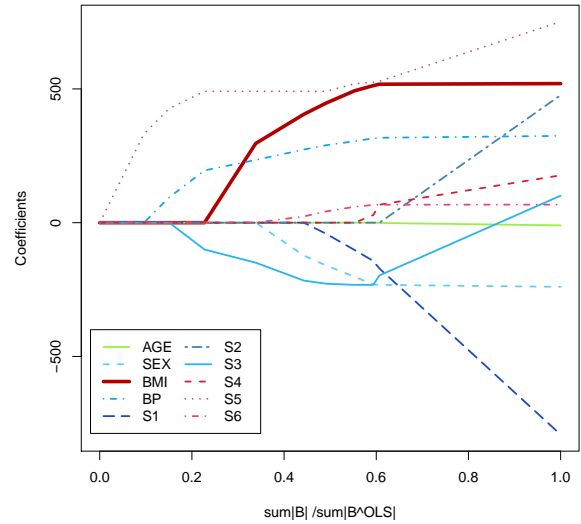
---

Le choix de la taille de la variable *step* est très simple puisque plus *step* est faible et plus le chemin de régularisation est précis. Cependant, le choix final de ce paramètre doit être un compromis entre la précision et les temps pour le calcul de ce modèle. On pourra déterminer automatiquement la valeur

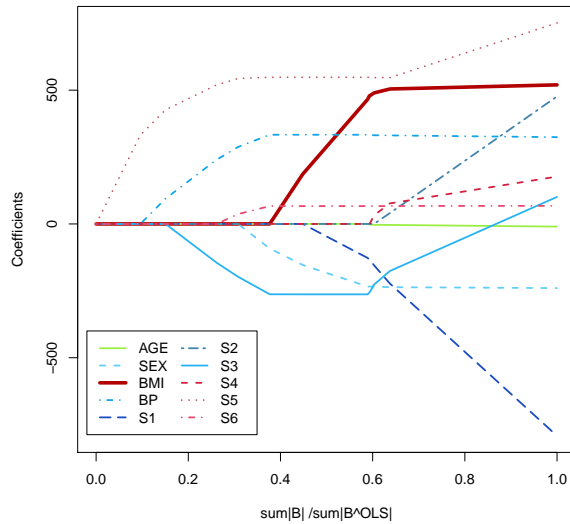
## 6. ADACOS



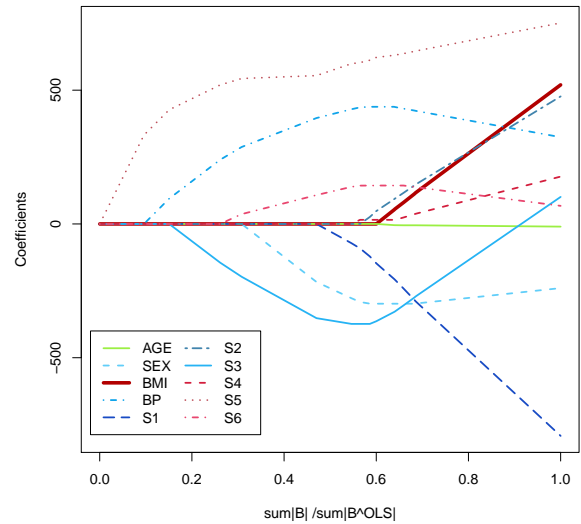
(a) Chemin de régularisation de la régression Forward AdaCOS avec  $\gamma = 0$



(b) Chemin de régularisation de la régression Forward AdaCOS avec  $\gamma = 1$



(c) Chemin de régularisation de la régression Forward AdaCOS avec  $\gamma = 2$



(d) Chemin de régularisation de la régression Forward AdaCOS avec  $\gamma = 6$

FIGURE 6.3 – Chemins de régularisation de la méthode Forward AdaCOS sur l'étude *Diabète* avec 4 valeurs de  $\gamma$  différentes :  $\gamma = \{0, 1, 2, 6\}$ . Pour visualiser le fonctionnement de cette méthode, seule la variable BMI est pénalisée :  $\alpha = \{0, 0, 1, 0, 0, 0, 0, 0, 0\}$ .

de ce paramètre si on ajuste un modèle OLS pour en déduire  $\sum_j |B_j^{OLS}|$ , l'amplitude maximale des coefficients sans pénalisation. De cette valeur sera déduite la taille optimale du paramètre *step*.

La section 6.4 propose une procédure qui permettra de généraliser cette méthode à toutes sortes de méthodes de classement.

## 6.4 Considérations pratiques et généralisation

Le calcul d'un modèle Forward AdaCOS est inefficace avec l'algorithme proposé dans la section 6.3. En effet, il est nécessaire de réaliser des centaines d'itérations afin de calculer le chemin de régularisation complet. On peut améliorer considérablement les temps de calcul en utilisant la méthode forward stagewise de l'algorithme LARS<sup>1</sup> d'Efron *et al.* [57].

### 6.4.1 La régression Forward AdaCOS avec LARS

Si nous reprenons la définition de base de la méthode AdaCOS :  $\hat{B}^{AdaCOS} = \operatorname{argmin}_B (\mathbf{y} - \mathbf{X}B)^T (\mathbf{y} - \mathbf{X}B) + \lambda B^T \mathbf{W}B$ , nous pouvons reformuler ce problème en normalisant la matrice  $\mathbf{X}$  différemment [257] :

$$\hat{B}^{*AdaCOS} = \operatorname{argmin}_B (\mathbf{y} - \mathbf{X}^*B)^T (\mathbf{y} - \mathbf{X}^*B) + \lambda B^T B, \quad (6.15)$$

en posant  $\mathbf{X}^* = \mathbf{X}/w$  et  $B^{*AdaCOS} = B^{AdaCOS} \times w$ . On obtient ainsi exactement la même formulation Ridge (voir l'équation 6.7) et par conséquent, les mêmes solutions. On peut donc utiliser le même procédé pour généraliser cette approche. En pondérant  $\mathbf{X}$  avec les poids du vecteur  $w$ , on pourra utiliser toutes les méthodes statistiques existantes. En utilisant l'algorithme LARS, nous obtenons donc une implémentation très rapide de la méthode Forward AdaCOS décrite dans la section précédente. Nous définissons par la même occasion les régressions LAR AdaCOS et LASSO AdaCOS qui fournissent sensiblement les mêmes résultats.

### 6.4.2 La régression logistique AdaCOS

Il est aussi possible d'étendre la méthode AdaCOS pour la régression logistique. La régression logistique [15, 138] fait partie des modèles linéaires multi-variables qui permettent de calculer sous forme de probabilités la relation existante entre les deux classes cibles et les variables prédictives correspondant à nos descripteurs. Cette méthode binomiale présente deux modalités 0 et 1. On note  $P(Y = 1)$  (resp.  $P(Y = 0)$ ), la probabilité à priori pour que  $Y = 1$  (resp.  $(Y = 0)$ ) et  $P(Y = 1|X)$  (resp.  $P(Y = 0|X)$ ), les probabilités à posteriori.

Le modèle Logit est défini de la manière suivante :

$$\ln \frac{P(Y = 1|X)}{1 - P(Y = 1|X)} = \mathbf{X}B, \quad (6.16)$$

on en déduit :

$$P(Y = 1|X) = \frac{e^{\mathbf{X}B}}{1 + e^{\mathbf{X}B}}. \quad (6.17)$$

Pour calculer cette probabilité, on utilise la méthode de descente de gradient de Friedman *et al.* [69] à travers le package R *glmnet*<sup>2</sup>.

---

1. R package Lars : <http://cran.r-project.org/web/packages/lars/>  
 2. R package glmnet : <http://cran.r-project.org/web/packages/glmnet/index.html>



Nous pouvons donc calculer des régressions logistiques Lasso AdaCOS, Ridge AdaCOS et Elastic net AdaCOS.

### 6.5 Résultats de classement

Nous proposons de montrer l'intérêt de la méthode AdaCOS à travers deux exemples d'applications : une sur la base de données *Diabète*, l'autre sur l'apprentissage statistique du projet TOCATA (voir l'annexe III).

#### 6.5.1 Base de données Diabète

On considère la problématique fictive suivante : pour maîtriser les coûts du contrôle de l'évolution du diabète, le prix maximal autorisé sera fixé et ne devra pas excéder une valeur donnée. On cherche donc un modèle précis et parcimonieux qui respecte cette contrainte de coût. Chaque descripteur aura donc un prix qui sera assigné arbitrairement :  $\alpha_{AGE} = 0$ ,  $\alpha_{SEX} = 0$ ,  $\alpha_{BP} = 30$ ,  $\alpha_{BMI} = 30$ ,  $\alpha_{S1} = 100$ ,  $\alpha_{S2} = 100$ ,  $\alpha_{S3} = 100$ ,  $\alpha_{S4} = 100$ ,  $\alpha_{S5} = 100$  et  $\alpha_{S6} = 100$ .

Le prix maximal pour réaliser un test est donc égal à 660€, si tous les descripteurs sont utilisés dans le modèle. On définit par  $A$ , le sous ensemble de descripteurs utilisés dans le modèle et par  $P_A$  le prix du test associé à ce sous ensemble ( $P_A = \sum_{j \in A} \alpha_j$ ). On impose la contrainte suivante  $P_A < 200\text{€}$  et on utilise **une régression linéaire LAR AdaCOS**.

#### 6.5.2 Résultats pour la base de données Diabète

Utiliser un modèle basé sur AdaCOS impose l'utilisation d'un second paramètre en plus de  $\lambda$  :  $\gamma$ . Ce paramètre contrôle l'influence de la fonction de coût à minimiser et il faut le déterminer automatiquement.

Pour mesurer la performance du classifieur et pour trouver les meilleurs valeurs pour  $\lambda$  et  $\gamma$ , on utilise une méthode de validation 2-D classique : la validation croisée *K-fold* [221]. Les individus sont divisés en  $K$  groupes de taille identique. L'apprentissage est réalisé sur  $K - 1$  groupes et ce modèle est validé sur le groupe écarté. Ce processus est répété  $K$  fois afin que chaque groupe serve d'ensemble de validation. On en déduit l'erreur quadratique moyenne (EQM) et son écart type (STD). Pour cette étude,  $K = 10$  et l'erreur quadratique est normalisée :  $(\mathbf{r}^T \mathbf{r}) / (\mathbf{y}^T \mathbf{y})$ . La figure 6.4(a) présente les résultats associés à cette validation croisée 2-D. On constate que l'erreur quadratique moyenne a tendance à augmenter lorsque  $\gamma$  augmente ce qui est conforme à nos attentes, puisque l'on biaise l'estimateur pour optimiser le prix global du test.

La figure 6.4(c) illustre l'écart type de l'EQM, qui ne semble pas être corrélé avec le paramètre  $\gamma$ . La grandeur qui nous intéresse plus particulièrement, le prix, est illustrée dans la figure 6.4(b). On observe l'effet recherché initialement, le prix global d'un test diminue lorsque  $\gamma$  augmente.

Pour  $\gamma = 0$ , on obtient exactement le chemin de régularisation de la régression LAR et on observe un phénomène intéressant : la régression LAR AdaCOS permet de prendre en compte le prix des descripteurs et propose, dans ce cas précis, un meilleur estimateur que la régression LAR puisque l'erreur quadratique passe de 0,519 à 0,499 soit un gain de 3,7%.

Lorsque l'on impose une contrainte de prix :  $P_A < 200\text{€}$ , (illustré par la figure 6.4(d) où la partie blanche de l'image ne respecte pas la contrainte de temps) alors ce gain est encore plus important et passe de 0,613 à 0,567 (gain de 7,5%). Le tableau 6.1 résume les paramètres sélectionnés et montre les descripteurs retenus. Pour cette base de données, il est possible de faire une recherche exhaustive du meilleur sous-ensemble de descripteurs. Avec 10 descripteurs ( $p = 10$ ), on compte 1023 sous-ensembles

TABLEAU 6.1 – Résultats pour la base de données *Diabète* avec une validation croisée *10-fold* (EQM, l'erreur quadratique moyenne, STD écart type de l'EQM,  $P_A$  prix global du test,  $\gamma$  et  $s$  les paramètres du modèle AdaCOS LAR).

Méthode	EQM	STD	$P_A$	$\gamma$	$s$	Descripteurs
LAR AdaCOS ( $P_A < 200$ )	0,567	0,111	160	20	0,37	AGE SEX BMI BP S5
LAR ( $P_A < 200$ )	0,613	0,068	160	0	0,25	AGE BMI BP S5
Exhaustive ( $P_A < 200$ )	0,520	0,088	160	-	-	AGE SEX BMI BP S5
LAR AdaCOS	0,499	0,087	660	28	0,80	AGE SEX BMI BP S1 S2 S3 S4 S5 S6
LAR	0,519	0,092	660	0	0,88	AGE SEX BMI BP S1 S2 S3 S4 S5 S6
Exhaustive	0,499	0,102	460	-	-	SEX BMI BP S1 S2 S4 S5

distincts et sur chacun d'entre eux, une régression linéaire non pénalisée est calculée. On intègre ces résultats aux précédents (tableau 6.1). On note que les résultats de la méthode exhaustive, bien que meilleurs, ne sont pas très éloignés de ceux des modèles LAR AdaCOS. Néanmoins, la recherche exhaustive du meilleur sous ensemble n'est possible que lorsque  $p < 20$ , rendant nécessaire les méthodes basées sur AdaCOS.

Un exemple concret d'utilisation de la méthode AdaCOS est donnée dans l'annexe III dans le cadre du projet TOCATA.

## 6.6 Bilan et perspectives

Nous avons introduit dans ce chapitre, une pénalisation différente pour répondre à de nouveaux types de problèmes : obtenir un modèle précis qui minimise en même temps une fonction de coût. Il s'agit donc de faire un compromis entre le biais introduit par les pénalisations adaptatives et la minimisation de la fonction de coût.

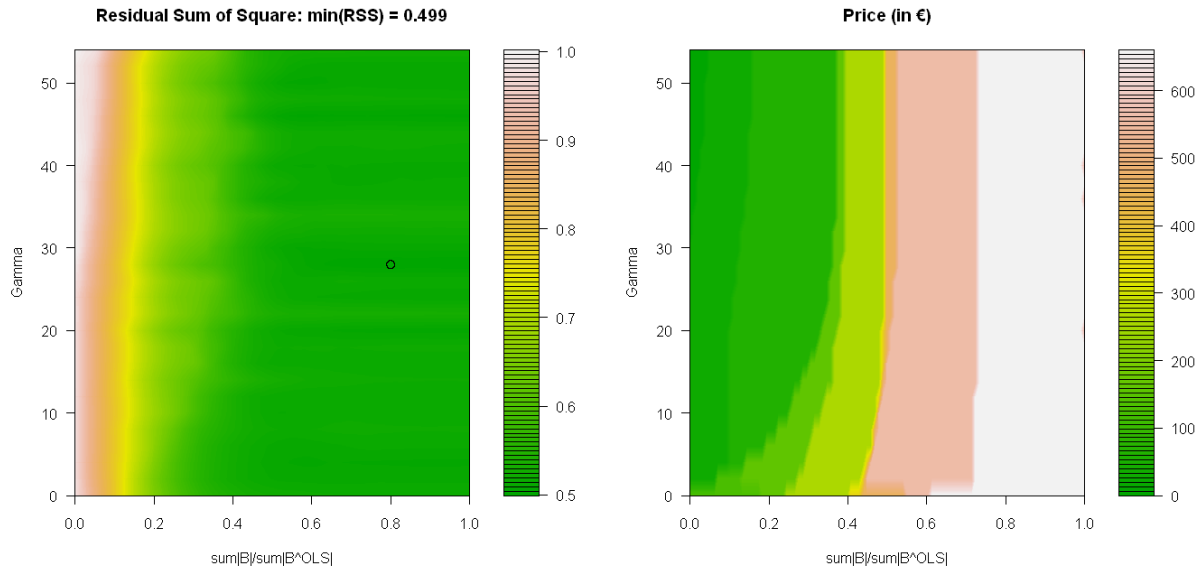
La régression AdaCOS pénalise différemment chaque descripteur et généralise par conséquent la régression Ridge, grâce au paramètre  $\gamma$ . Ce paramètre permet de changer d'une manière continue l'influence de la fonction de coût (un compromis est à trouver entre cette fonction et la précision de la régression).

La méthode AdaCOS prend tout son sens lorsque l'on construit des modèles parcimonieux. Ainsi, une sélection de variables peut être réalisée et le *coût global* d'un modèle peut être réduit. On propose un algorithme glouton pour étendre la fonction forward stagewise puis on développe une méthodologie pour utiliser tous les modèles existants avec la méthode AdaCOS. Les temps de calcul sont donc fortement réduits par rapport à une approche gloutonne, puisque les chemins de régularisation sont calculés analytiquement avec l'algorithme LARS. On pourra donc aussi calculer des régressions logistiques parcimonieuses AdaCOS.

On présente les résultats (annexe III) pour le classement pixel à pixel de défauts de surface et on montre qu'un modèle AdaCOS permet de réduire l'erreur quadratique moyenne.

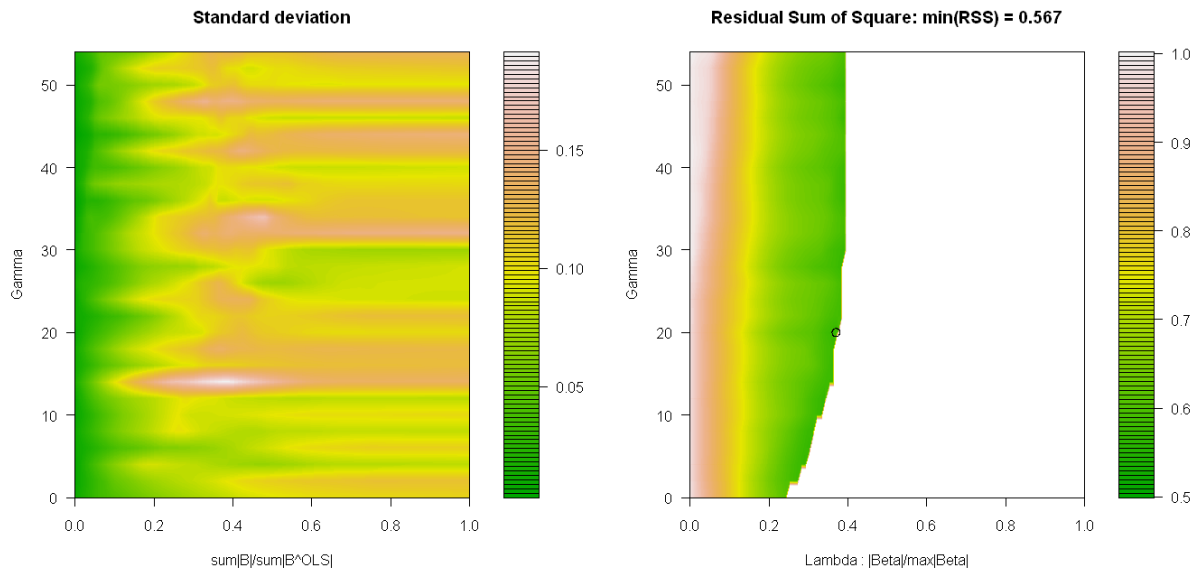
Comme travail futur, nous souhaitons faire une étude spécifique sur l'influence du bruit dans un modèle AdaCOS. La preuve d'un bon rapport signal sur bruit facilitera l'adoption de cette méthode par la communauté statistique. De plus, nous travaillons sur une méthode axée sur la méthodologie AdaCOS pour obtenir un modèle parcimonieux par groupe [139] qui prend en compte une fonction de coût.

## 6. ADACOS



(a) Erreur quadratique normalisée. Le cercle représente la solution retenue par validation croisée *10-fold*.

(b) Coût du test



(c) Ecart type de l'erreur quadratique normalisée

(d) Erreur quadratique normalisée corrigée par la contrainte de coût  $P_A < 200\text{€}$ . Le cercle représente la solution retenue par validation croisée *10-fold*.

FIGURE 6.4 – Résultats de la validation croisée *10-fold* pour l'étude *diabète*. (a) L'erreur quadratique moyenne normalisée, (b) le coût du test en fonction des descripteurs sélectionnés dans le modèle, (c) l'écart type de l'erreur quadratique normalisée et (d) l'erreur quadratique normalisée avec la contrainte  $P_A < 200\text{€}$ . Les pénalisations utilisées sont  $\alpha_{AGE} = 0$ ,  $\alpha_{SEX} = 0$ ,  $\alpha_{BP} = 30$ ,  $\alpha_{BMI} = 30$ ,  $\alpha_{S1} = 100$ ,  $\alpha_{S2} = 100$ ,  $\alpha_{S3} = 100$ ,  $\alpha_{S4} = 100$ ,  $\alpha_{S5} = 100$  et  $\alpha_{S6} = 100$ .

Deuxième partie

**Applications industrielles**



## CHAPITRE 7

---

Projet TOCATA : Extraction automatique des défauts de surface de pièces métalliques par traitement d'images et analyse statistique

---

*Chapitre confidentiel*

## 7. PROJET TOCATA

---

---

### Projet L'Oréal : Caractérisation 3-D *in vivo* du milieu fibreux du derme de la peau humaine par traitement d'images

---

La microscopie multiphoton permet d'obtenir des images en trois dimensions (3-D) particulièrement adaptées à l'analyse de la peau humaine. Les acquisitions sont réalisées *in vivo* pour quantifier les structures de la peau sans avoir besoin de faire une biopsie (prélèvement d'un échantillon de peau du patient). Il s'agit donc de faire un contrôle non invasif et non destructif. Les images obtenues ont une résolution spatiale d'une centaine de micromètres permettant de visualiser l'épiderme ainsi que la partie supérieure du derme avec les fibres de collagène et d'élastine. Ces réseaux de fibres procurent le tonus et l'élasticité à la peau et comprendre leur évolution dans le temps est un élément essentiel dans un contexte médical et cosmétique.

Dans ce chapitre, on propose différents descripteurs morphologiques pour caractériser la forme, l'organisation et la distribution de tailles des réseaux de fibres de la partie supérieure du derme de la peau. Les outils de détections des éléments fins développés dans ce document sont utilisés dans cette application pour mieux comprendre leur évolution avec le temps.

Une analyse statistique complète cette étude afin de définir quels sont les descripteurs les plus pertinents pour faire la distinction entre une peau jeune et une peau âgée. À notre connaissance, cette étude est la première pour la caractérisation en trois dimension du réseau de fibres de la partie supérieure du derme de la peau humaine.

Ce projet est réalisé en partenariat entre le CMM et le groupe industriel français de produits cosmétiques **L'Oréal**<sup>1</sup>.

Mots clés : Caractérisation de la peau, traitement d'images 3-D, morphologie mathématique, imagerie multiphoton, caractérisation des fibres du derme, analyse statistique, test d'hypothèses.

---

1. L'Oréal R&D - Aulnay, France : [www.loreal.fr](http://www.loreal.fr)



Plusieurs publications, communications et brevets ont été réalisés pour présenter et protéger les résultats obtenus dans le cadre du partenariat entre L'Oréal et le Centre de Morphologie Mathématique.

1. Dans Morard *et al.* [154], quelques descripteurs pour caractériser le réseaux de fibres sont présentés à la conférence nationale de stéréologie ISS France, puis au workshop international 3DMM [153].

**V. Morard, E. Decencièrre, P. Dokládal, Méthodes géodésiques pour la caractérisation des milieux fibreux, 34ème journée ISS France, Paris, France, 2011**

**V. Morard, E. Decencièrre, P. Dokládal, Characterization of 3-D fibrous media with geodesic methods, 3-D Microstructure Meeting (3DMM), Saarbrücken, Germany, 2011**

2. Des résultats plus généraux ont été présentés dans Baldeweck *et al.* [12] à la conférence internationale Photonics West puis dans [11].

**T. Baldeweck, E. Tancrède-Bohin, P. Dokládal, S. Koudoro, V. Morard, F. Meyer, E. Decencièrre, A.M. Pena, In vivo multiphoton microscopy associated to 3-D image processing for human skin characterization, Multiphoton Microscopy in the Biomedical Sciences, Photonics West, San Francisco, USA, 2012**

**T. Baldeweck, E. Tancrède-Bohin, P. Dokládal, S. Koudoro, V. Morard, F. Meyer, E. Decencièrre, A.M. Pena, Imagerie multiphoton de la peau humaine in vivo, Diagnostic et imagerie optique en médecine (OPTDIAG), Paris, France, 2012**

3. Enfin, un brevet a été déposé dans le cadre de cette collaboration [9] :

**T. Baldeweck, A.M. Pena, E. Tancrède-Bohin, E. Decencièrre, P. Dokládal, S. Koudoro, V. Morard, F. Meyer, Procédé pour caractériser l'épiderme et le derme à partir d'images multiphoton tridimensionnelles in vivo de la peau, numéro 1161616, 14 décembre 2011**

### 8.1 Présentation du projet L'Oréal

Le vieillissement de la peau est sans doute un des éléments les plus visibles des effets de l'âge. Il fait intervenir des processus complexes qui reposent sur des facteurs intrinsèques (génétiques, hormonaux, maladies) et extrinsèques (l'environnement). L'un des facteurs externes principaux favorisant le vieillissement de la peau est l'exposition répétée au soleil ; on parle alors de *photovieillissement* ou d'*héliodermie*. Ce phénomène ne sera pas étudié dans ce document puisque les acquisitions seront réalisées sur des zones de la peau peu exposées au soleil : la partie ventrale de l'avant bras.

Le vieillissement de la peau se traduit par l'atrophie du derme et de l'épiderme, par la sécheresse, par la réduction de l'élasticité et de la force de la peau, par l'apparition de ridules ou par la fragilisation

des vaisseaux sanguins. D'un point de vue histologique, on mesure une diminution de l'épaisseur de l'épiderme [28] et un aplatissement de la jonction dermo-épidermique (JDE) [112].

### 8.1.1 Acquisition des images



FIGURE 8.1 – Système optique

La microscopie multiphoton est une technique d'imagerie qui utilise une excitation des fluorophores d'un échantillon, avec deux ou trois photons. Ce système optique est devenu au cours des dix dernières années reconnu pour ses nombreuses qualités : c'est une technique non invasive, qui génère des informations en trois dimensions, possédant une résolution inférieure au micromètre et avec plusieurs modalités complémentaires : l'auto-fluorescence (AF) et la génération de second harmonique (SHG). Ce système a depuis été utilisé pour des études sur des peaux normales [109], sur l'effet de l'âge et du soleil [110], sur les mélanomes [53, 176], sur les fibres de collagène [2] et pour des recherches pharmaceutiques [111].

Le signal AF est émis par certaines structures de la peau comme la mélanine, la kératine ou l'élastine alors que le signal SHG est fort pour des structures denses macromoléculaires comme les fibres de collagène [133, 256, 181].

Le système optique permettant de réaliser l'acquisition des signaux AF et SHG est composé du *DermaInspect®* (JenLab) avec le système FLIM décrit dans [58, 10] et avec un laser femtoseconde Titanium-Sapphire (MaiTai Spectra-Physics) de longueur d'onde 760 nm.

### 8.1.2 Base de données expérimentale

L'Oréal a mis au point une campagne d'acquisition à l'hôpital Saint Louis<sup>1</sup>, Paris [12], pour visualiser les peaux (avant bras, position ventrale) de 15 personnes jeunes (18-25 ans) et de 15 personnes âgées (70-75 ans). Deux échantillons ont été réalisés sur chaque volontaire, portant à 60 le nombre d'images labellisées en deux classes : jeune et âgée. Tous les volontaires sont des femmes ayant un phototype de Fitzpatrick compris entre I et IV (peau claire) pour éviter les variabilités introduites par le sexe ou des peaux plus mates.

### 8.1.3 Objectifs de l'étude

L'objectif de cette étude consiste à concevoir une méthodologie et un cadre de travail adaptés pour caractériser les réseaux de fibres de la partie supérieure du derme de la peau humaine afin de mieux connaître leurs caractéristiques morphologiques. Savoir comment ils évoluent avec le temps permet de comprendre les modifications structurelles des fibres de la peau, élément indispensable pour mesurer scientifiquement l'apport de crèmes anti-âge.

Les fibres de collagène et d'élastine jouent un rôle important sur le tonus et l'élasticité de la peau. Réaliser une biopsie d'un échantillon de peau a pour conséquence de modifier sa tension surfacique, et ainsi, de modifier les caractéristiques morphologiques des fibres. Il est donc indispensable d'utiliser un système d'imagerie **in-vivo** tel que la microscopie multiphoton.

Avoir des images en trois dimensions apporte des informations essentielles sur les structures de la peau. Les fibres sont elles aussi en trois dimensions. Par conséquent, la caractérisation du réseau de fibres sera réalisée exclusivement en **trois dimensions**.

1. <http://ghparis10.aphp.fr/>

On rappelle que la peau est composée de deux couches distinctes : l'épiderme en surface et le derme en profondeur. Dans un premier temps, une **segmentation 3-D** est indispensable pour isoler toutes les structures du derme de la peau. Or, la forme de la jonction entre le derme et l'épiderme est complexe et cette segmentation devra utiliser des méthodes de segmentation complexes. Dans un second temps, un **filtre linéaire** sera appliqué pour atténuer le bruit et pour supprimer les structures des images qui ne sont pas des fibres.

Une étude sera menée sur l'orientation, la forme, la distribution de tailles et sur l'organisation du réseau de fibres, suivie d'un test de significativité.

### 8.1.4 Organisation du chapitre

La section 8.2 présente les prétraitements nécessaires pour isoler et rehausser les fibres des images 3-D. Suit l'introduction des différentes régions d'intérêt du derme (section 8.2.3). La section 8.3 propose différents descripteurs morphologiques pour caractériser la forme, la taille, l'orientation ou l'organisation spatiale des fibres. Enfin, une analyse statistique (section 8.4) permet de visualiser les paramètres qui évoluent avec le temps afin de soumettre une interprétation possible des résultats (section 8.5).

## 8.2 Prétraitements

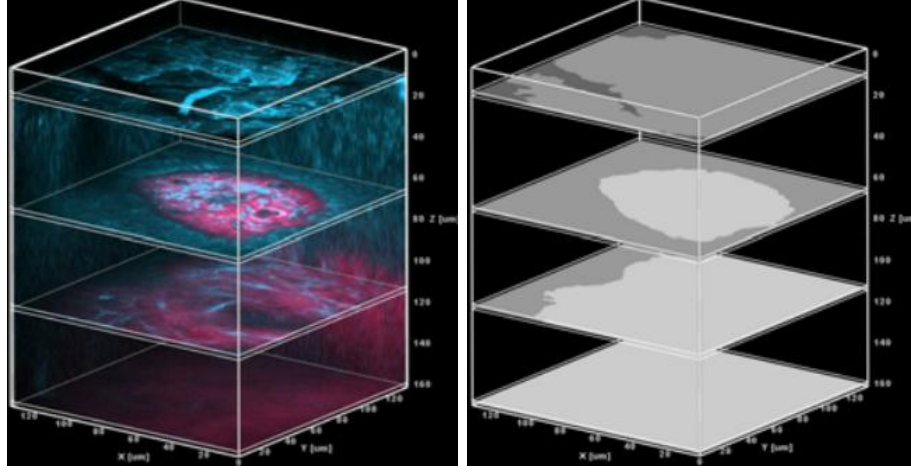
L'image d'entrée  $f$  possède deux modalités appelées AF et SHG,  $f = (f^{AF}, f^{SHG})$ . Chaque canal est une image 3-D telle que  $f^{AF} : D \rightarrow V$  et  $f^{SHG} : D \rightarrow V$ . Le support de l'image  $D \subset \mathbb{Z}^3$  est une grille cubique de taille  $511 \times 511 \times 70$  dans les directions  $x, y$ , et  $z$  respectivement et  $V$  est l'ensemble des valeurs entières et non négatives avec une précision de 12 bits. Le plan formé par les axes  $x-y$  est coplanaire à la surface de la peau et des valeurs croissantes sur l'axe  $z$  signifient que la profondeur dans la peau augmente. Les images sont acquises avec une forte anisotropie dans l'axe  $z$  puisque la résolution spatiale est de  $0,255 \mu m$  pour les axes  $x$  et  $y$  et de  $2,346 \mu m$  pour l'axe  $z$ . On observe donc un volume de  $130,4 \times 130,4 \times 164,2 \mu m^3$  ce qui permet d'observer les structures dans l'épiderme mais aussi dans le derme. Dans cette étude, seules les structures fibreuses du derme nous intéressent et pour les isoler, il est nécessaire de segmenter les images.

### 8.2.1 Segmentation 3-D : air/épiderme/derme

Pour identifier la région d'intérêt et pour caractériser uniquement les structures fibreuses, une segmentation est nécessaire pour trouver les séparations entre la surface et l'épiderme puis entre l'épiderme et le derme. Dans [49], Decencièr *et al.* proposent une méthode 3-D, automatique, basée sur la ligne de partage des eaux [21] suivie d'une approche par coupes de graphes [121, 217, 79]. Cette segmentation nous donne les masques de l'air, du derme et de l'épiderme pour chaque image 3-D et la figure 8.2 illustre les résultats obtenus. Une fois la segmentation calculée, on peut donc facilement isoler toutes les structures de l'image appartenant au derme.

### 8.2.2 Rehaussement des fibres en 3-D

On cherche à quantifier le réseau fibreux du derme de la peau, aussi, il faut supprimer les structures du derme qui ne présentent pas d'intérêt. Nous concevons un filtre linéaire pour réaliser un rehaussement 3-D des fibres de l'image, basé sur la différence de deux gaussiennes (DoG). Soit  $h : \mathbb{Z}^3 \rightarrow \mathbb{R}$  la réponse



(a) Image brute en 3-D

(b) Segmentation Air/Epiderme/Derme

FIGURE 8.2 – Illustration des résultats de la segmentation entre trois phases sur une image 3-D : en gris foncé est représenté l'air, en gris l'épiderme et en gris clair le derme de la peau (sujet jeune).

impulsionnelle du filtre et  $H : \mathbb{Z}^3 \rightarrow \mathbb{C}$  sa réponse fréquentielle avec  $H = \mathcal{F}(h)$ ,  $\mathcal{F}$  étant la transformée de Fourier [65, 25].

### Création du filtre

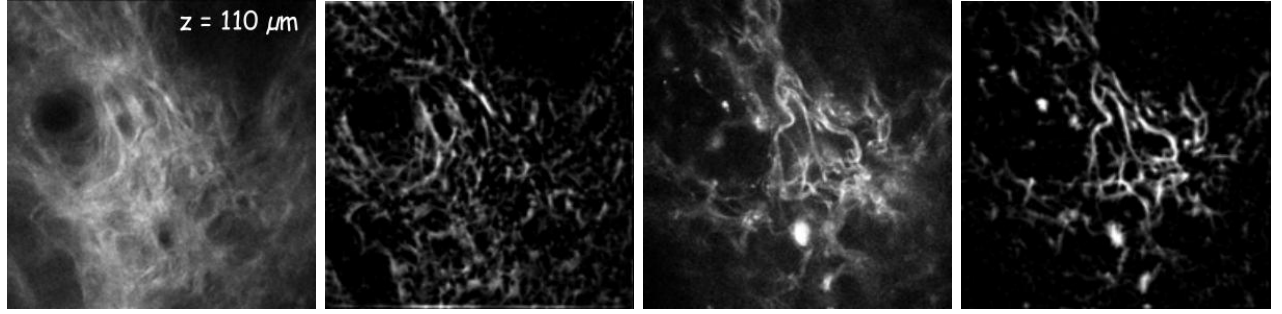
La forme du noyau sera déterminée à partir de la forme géométrique des structures à rehausser. Pour des fibres,  $h$  est un noyau 3-D qui aura un axe principal  $z$  et deux axes secondaires notés  $x$  et  $y$ .

Soit  $h_x$ ,  $h_y$  et  $h_z$  des sections centrées de  $h$  le long des axes  $x$ ,  $y$  et  $z$  respectivement. Le noyau est symétrique autour de l'axe principal  $z$  donnant  $h_x = h_y$ . De plus, les sections perpendiculaires à l'axe  $z$  ( $h_x$  et  $h_y$ ) ont un profil laplacien, obtenu par la soustraction de deux gaussiennes (*Difference of Gaussians*)  $G_\sigma - G_{2\sigma}$ . Cet écart type  $\sigma$  est un paramètre déduit de la largeur des fibres à rehausser, soit  $0,765 \mu\text{m}$  (correspondant à des fibres de 3 pixels de largeur). Enfin, la section  $h_z$  a un profil gaussien  $G_\sigma$  avec  $\sigma$  fixé à  $2,5 \mu\text{m}$ . Cela permet de détecter des fibres de courbure faible et pouvant être considérées comme localement rectiligne sur  $5 \mu\text{m}$ .

Le gain de la réponse fréquentielle de ce noyau est normalisé à 1 en posant  $H' = H/\max|H|$ . La moyenne de  $h'$  est donc égale à 0 et par conséquent, ce filtre atténuera les zones plates puisque  $H'(0) = 0$ .

### Filtrage

Pour rehausser les fibres orientées suivant  $\psi$ , avec  $\psi$  une orientation 3-D comprenant l'élévation et l'azimut ( $\psi = (\theta, \phi)$ ), on prend  $h'_\psi$ , une copie de  $h'$  ayant subi une rotation suivant  $\psi$ . Ainsi, le filtre  $h'_\psi$  donne une réponse maximale pour des objets qui ont une géométrie proche de celle de  $h$  et orientés suivant une des directions  $\psi_i \in \Psi$ . On discrétise ce noyau dans  $D$  en prenant en compte les pas d'échantillonnage  $d_x$ ,  $d_y$  et  $d_z$  de l'image. Enfin, pour détecter les fibres dans toutes les orientations de l'espace, on fait un échantillonnage angulaire sur les directions  $\Psi = \{\psi_i\}$  et on génère un banc de filtres directionnels  $h^\Psi$ . On rappelle qu'une convolution dans le domaine spatial correspond à une



(a) Fibres de collagène (b) Fibres rehaussées de (a) (c) Fibres élastiques (AF) (d) Fibres rehaussées de (c) (SHG)

FIGURE 8.3 – Rehaussement des fibres du derme de la peau d'un sujet jeune sur les modalités AF et SHG.

multiplication point à point dans le domaine fréquentiel, d'où le filtrage de  $f$  par  $H_\Psi$  :

$$f^{RF3D} = \bigvee_{\psi_i \in \Psi} \mathcal{F}^{-1}(H'_{\psi_i} \mathcal{F}(f)), \quad (8.1)$$

avec  $\mathcal{F}^{-1}$  la transformée de Fourier inverse. L'image  $f^{RF3D}$  contient les fibres rehaussées (les fibres corrompues par du bruit ont été reconnectées). De plus, le rapport signal sur bruit a été considérablement augmenté, comme l'atteste les images de la figure 8.3.

### 8.2.3 Les régions d'intérêt

Nous disposons des fibres rehaussées de la peau mais elles n'occupent pas tout le volume du derme. On définit dans cette section le masque binaire des fibres permettant de savoir où elles sont situées, ainsi que les différentes régions d'intérêt appelées tranches, afin d'affiner la quantification.

#### Le masque des fibres (MF)

Les fibres sont interconnectées et tortueuses, formant un réseau dense. Leur extraction individuelle est une tâche difficile qui ne sera pas entreprise dans cette étude. Par conséquent, on extrait un masque binaire du réseau fibreux en réalisant une segmentation de l'image filtrée  $f^{RF3D}$ . L'intensité du signal, provenant du microscope, décroît exponentiellement avec la profondeur du signal dans la peau (dans la direction  $z$ ). Aussi, on utilise un seuillage adaptatif avec la profondeur de l'image pour compenser cette décroissance du signal.

Le masque du réseau fibreux est noté  $f^{MF}$  :

$$f^{MF} = \{(x,y,z) \mid f^{RF3D}(x,y,z) \geq \underset{\forall i,j}{\text{mean}}(f^{RF3D}(i,j,z)) + 2 \underset{\forall i,j}{\text{std}}(f^{RF3D}(i,j,z))\}, \quad (8.2)$$

avec les fonctions  $\text{mean}()$  et  $\text{std}()$  respectivement la valeur moyenne et l'écart type. Cette segmentation adaptative fournit un masque des fibres restrictif puisque seules les fibres les plus contrastées sont marquées.

### Les tranches : des régions d'intérêt adaptées

Il peut être intéressant de définir des régions d'intérêt plus petites et localisées dans certaines zones de l'image. Les zones intéressant particulièrement les biologistes se situent sous la jonction dermo-épidermique. À partir de la segmentation obtenue (section 8.2.1), la forme de la JDE est connue et il est alors possible de calculer une carte de distances pour qu'en chaque point du derme, on obtienne la distance minimale à la JDE. Les lignes de niveaux de cette carte de distances permettent de définir 3 tranches dans le derme plus ou moins éloignées de cette jonction.

Ainsi, le derme total ( $DT$ ) est la première région d'intérêt et elle englobe tout le derme présent dans l'image. Les tranches  $Tr1$  et  $Tr2$  sont des tranches de  $20 \mu m$  d'épaisseur qui sont situées respectivement à 0 et  $20 \mu m$  de la JDE. La dernière tranche ( $Tr3$ ) occupe le reste de la partie supérieure du derme :  $DT \setminus \{Tr1 \cup Tr2\}$ . On rappelle que le derme peut atteindre  $1 mm$  d'épaisseur et qu'une grande partie du derme reste invisible avec cette technique d'acquisition *in vivo*. L'utilisation de ces régions d'intérêt nécessite de considérer uniquement les structures présentes dans ces régions et d'ignorer celles qui sont à l'extérieur.

## 8.3 Caractérisation du réseau de fibres

L'objectif de cette section est de trouver des mesures pour la caractérisation du réseau de fibres de la partie supérieure du derme de la peau humaine. Chaque mesure sera globale, représentée par une valeur scalaire calculée sur une image 3-D. On utilise ainsi des méthodes de traitement d'images pour décrire l'orientation, la forme (longueur, allongement, tortuosité) ou l'organisation spatiale des fibres.

Chaque descripteur caractérisant ce réseau sera calculé indépendamment sur toutes les régions d'intérêt de l'image et sur les deux modalités AF et SHG. Ainsi, un descripteur est noté :  $Nom-du-descripteur(mod,roi)$  avec  $mod \in \{AF, SHG\}$  et  $roi \in \{DT, Tr1, Tr2, Tr3\}$ .

### 8.3.1 Orientation du réseau de fibres

L'orientation locale voxel à voxel des fibres est déduite des ouvertures directionnelles définies par l'équation 2.5 dans le chapitre 2. On définit la transformation  $\zeta : D \rightarrow \Psi$ , où l'on assigne à chaque voxel, la valeur de l'angle préférentiel parmi l'ensemble  $\Psi$  :

$$\zeta(f^{RF3D}) = \operatorname{argmax}_{\psi_i \in \Psi} \gamma_L^{\psi_i}(f^{RF3D}), \quad (8.3)$$

avec  $L$ , la longueur du segment de droite fixé à  $5 \mu m$ . On considère une région d'intérêt  $E \subset D$ . L'estimation de l'orientation préférentielle des fibres dans  $E$  ainsi que l'isotropie  $A$  passe par l'analyse de la carte d'orientations  $\zeta$ . On note  $\zeta_x$ ,  $\zeta_y$  et  $\zeta_z$  les composantes du vecteur d'orientation le long des axes  $x$ ,  $y$  et  $z$  respectivement. On construit alors le tenseur d'orientation de second ordre :

$$J = \begin{pmatrix} \zeta_x^2 & \zeta_x \zeta_y & \zeta_x \zeta_z \\ \zeta_y \zeta_x & \zeta_y^2 & \zeta_y \zeta_z \\ \zeta_z \zeta_x & \zeta_z \zeta_y & \zeta_z^2 \end{pmatrix}$$

avec  $\zeta_i \zeta_j$ , les moments de second ordre :

$$\zeta_i \zeta_j = \sum_{x \in E} \zeta_i(x) \zeta_j(x) f^{RF3D}(x) \quad (8.4)$$

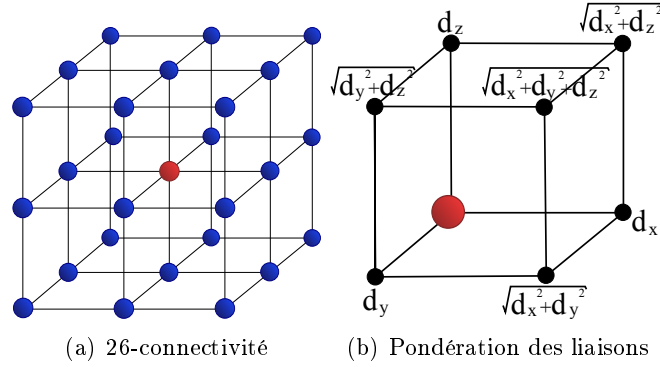


FIGURE 8.4 – Illustrations de la connectivité 3-D et de la pondération utilisées dans cette étude (voir la section 8.2 pour les valeurs de  $d_x$ ,  $d_y$  et  $d_z$ ).

On réalise la décomposition en valeurs singulières de la matrice  $J$  puis on trie les vecteurs propres par ordre décroissant des valeurs propres associées  $\lambda_i$ ,  $i = \{1, 2, 3\}$ ,  $\lambda_1 \geq \lambda_2 \geq \lambda_3$  pour déterminer le vecteur propre qui a le plus d'importance,  $u_1$ . Ainsi, ce vecteur propre  $u_1$  est orienté suivant l'orientation préférentielle des fibres dans  $E$ .

Pour un milieu isotrope (sans orientation privilégiée) et anisotrope (orienté), les valeurs propres  $\lambda_i$  prennent les valeurs données dans le tableau 8.1.

TABLEAU 8.1 – Valeurs propres pour des distributions isotropes et anisotropes.

$\lambda_i$	1	2	3
isotrope	1	1	1
anisotrope	1	0	0

On peut donc définir une mesure d'anisotropie pour une distribution de fibres que l'on note  $A$  avec  $A \in [0, 1]$  :

$$A = 1 - \frac{\lambda_2 + \lambda_3}{2\lambda_1} \quad (8.5)$$

Si  $A = 1$ , le milieu est anisotrope et les fibres sont orientées suivant  $u_1$  et si  $A = 0$ , alors il n'y a pas d'orientation préférentielle (milieu est isotrope) et le vecteur propre  $u_1$  n'est pas significatif.  $A$  et  $u_1$  sont donc deux descripteurs que l'on notera  $AnisXYZ(mod, roi)$  (resp.  $AnisJDE(mod, roi)$ ) et  $PhiXYZ(mod, roi)$  (resp.  $PhiJDE(mod, roi)$ ) pour l'anisotropie et l'orientation principale dans le référentiel xyz (resp. par rapport à la JDE).

### 8.3.2 Longueur apparente des fibres

La caractérisation de la longueur du réseau de fibres peut être réalisée avec les méthodes géodésiques (introduites dans le chapitre 5). Ainsi, le diamètre barycentrique est utilisé pour mesurer la longueur d'un objet et caractériser globalement la longueur moyenne des objets de l'image 3-D.

On utilise le masque des fibres, associé à une 26-connectivité (voir la figure 8.4(a)). Pour prendre en compte l'anisotropie du signal 3-D, chaque liaison sera pondérée par sa distance au voxel central et la figure 8.4(b) résume les poids utilisés pour cette connectivité.

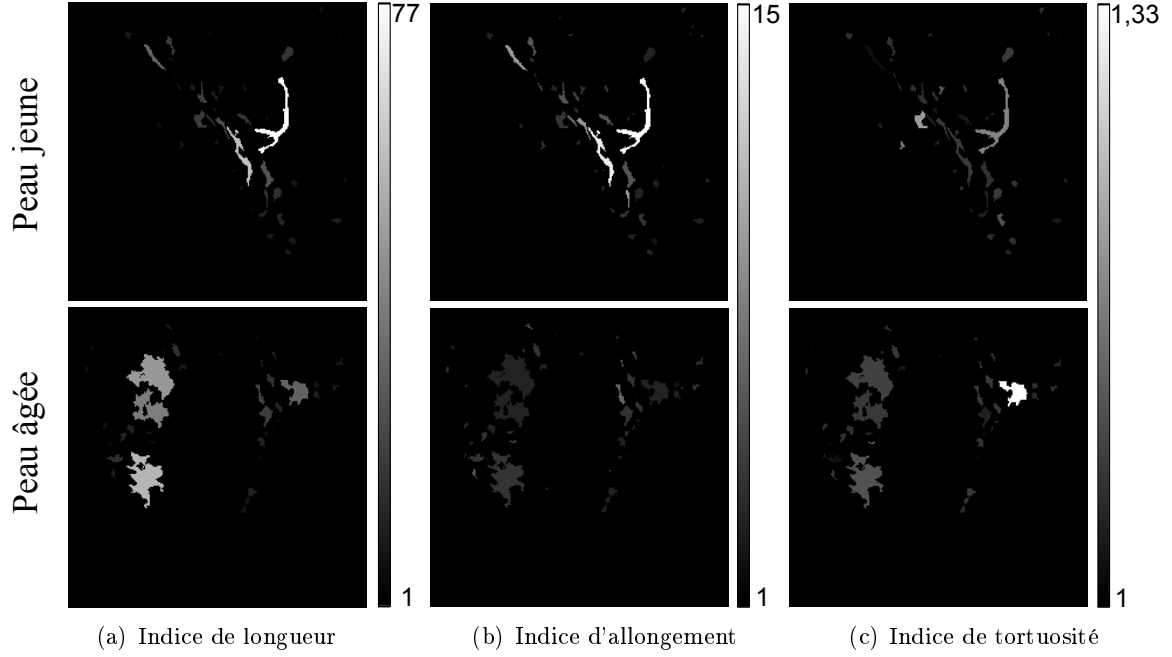


FIGURE 8.5 – Illustration des résultats des opérateurs (a)  $\zeta^{\text{length}}$ , (b)  $\zeta^{\text{elongation}}$  et (c)  $\zeta^{\text{tortuosity}}$  sur deux coupes 2-D extraites de deux images 3-D différentes.

Le descripteur associé à cette mesure calcule la longueur moyenne des objets pondérée par leur taille. Par objet, on entend une composante connexe du réseau de fibres. Si l'image est composée de 3 objets ayant un volume  $(v_1, v_2, v_3)$  et une longueur  $(l_1, l_2, l_3)$ , alors la valeur du descripteur sera égale à  $\frac{v_1 l_1 + v_2 l_2 + v_3 l_3}{v_1 + v_2 + v_3}$ . Pour définir ce descripteur noté  $Length(mod, roi)$ , on définit tout d'abord la transformation  $\xi^{\text{length}} : D \rightarrow \mathbb{N}$  :

$$\xi^{\text{length}}(f)(x) = Lg(X), x \in X, \quad (8.6)$$

un opérateur transformant une image binaire en une image en niveaux de gris, où chaque voxel est valué par le diamètre barycentrique de l'objet auquel il appartient. Le descripteur  $Length(mod, roi)$  est alors noté formellement :

$$Length(mod, roi) = \frac{\sum_{x \in roi} (\xi^{\text{length}}(f^{MF, mod}))(x)}{\sum_{x \in roi} f^{MF, mod}(x)}, \quad (8.7)$$

avec  $f^{MF, mod}$  le masque des fibres binaire associé au canal AF ou SHG. Ce descripteur permet de calculer la longueur moyenne pondérée des objets de l'image.

### 8.3.3 Allongement apparent des fibres

Avec le diamètre barycentrique, seule la longueur des structures est analysée. L'indice d'allongement d'un objet quantifie son élongation. Pour une sphère, son indice d'allongement est égal à 1 et plus une structure est fine et longue et plus cet indice est important (voir [116] ou voir la section 5.3.2, équation



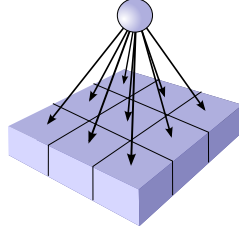


FIGURE 8.6 – Graphe d'adjacence 3-D utilisé pour construire des ouvertures parcimonieuses par chemins pour la direction N-S. Pour d'autres orientations, on applique des rotations à ce graphe.

5.9). De la même manière que le descripteur de longueur, on mesure l'indice d'allongement moyen pondéré sur une image :

$$Elong(mod,roi) = \frac{\sum_{x \in roi} (\zeta^{elongation}(f^{MF,mod}))(x)}{\sum_{x \in roi} f^{MF,mod}(x)}, \quad (8.8)$$

avec  $\zeta^{elongation}$ , un opérateur valant les voxels d'un objet par la valeur de l'indice d'allongement de l'objet auquel il appartient.

### 8.3.4 Tortuosité apparente des fibres

De la même manière que pour la longueur et l'allongement des fibres, on peut utiliser l'indice de tortuosité défini dans le chapitre 5.3.3 pour caractériser les fibres d'une image. Le descripteur associé à la tortuosité moyenne pondérée est écrit :

$$Tort(mod,roi) = \frac{\sum_{x \in roi} (\zeta^{tortuosity}(f^{MF,mod}))(x)}{\sum_{x \in roi} f^{MF,mod}(x)}, \quad (8.9)$$

avec  $\zeta^{tortuosity}$ , un opérateur valant les voxels d'un objet par la valeur de l'indice de tortuosité d'un objet auquel il appartient.

La figure 8.5 présente les résultats des opérateurs  $\zeta^{length}$ ,  $\zeta^{elongation}$  et  $\zeta^{tortuosity}$  sur des coupes 2-D de fibres pour une peau jeune (première ligne) ou pour une peau âgée (seconde ligne). On remarque que toutes les structures longues sont rehaussées pour un indice de longueur (image 8.5(a)), cependant, seules les structures filiformes le sont avec un indice d'allongement (figure 8.5(b)).

### 8.3.5 Distribution de tailles apparente des fibres

Les trois précédents descripteurs sont basés sur des méthodes géodésiques. Ils utilisent une 26-connexité et analysent le masque des fibres, une image binaire. On propose ici de restreindre la connexité des structures en utilisant des ouvertures parcimonieuses par chemins (PPO) en 3-D sur les fibres en niveaux de gris.

### Extension des PPO aux images 3-D

Une ouverture parcimonieuse par chemins est un nouvel opérateur qui a été présenté dans le chapitre 3 de cette thèse pour des images 2-D. L'extension de cet opérateur à des images 3-D est simple et

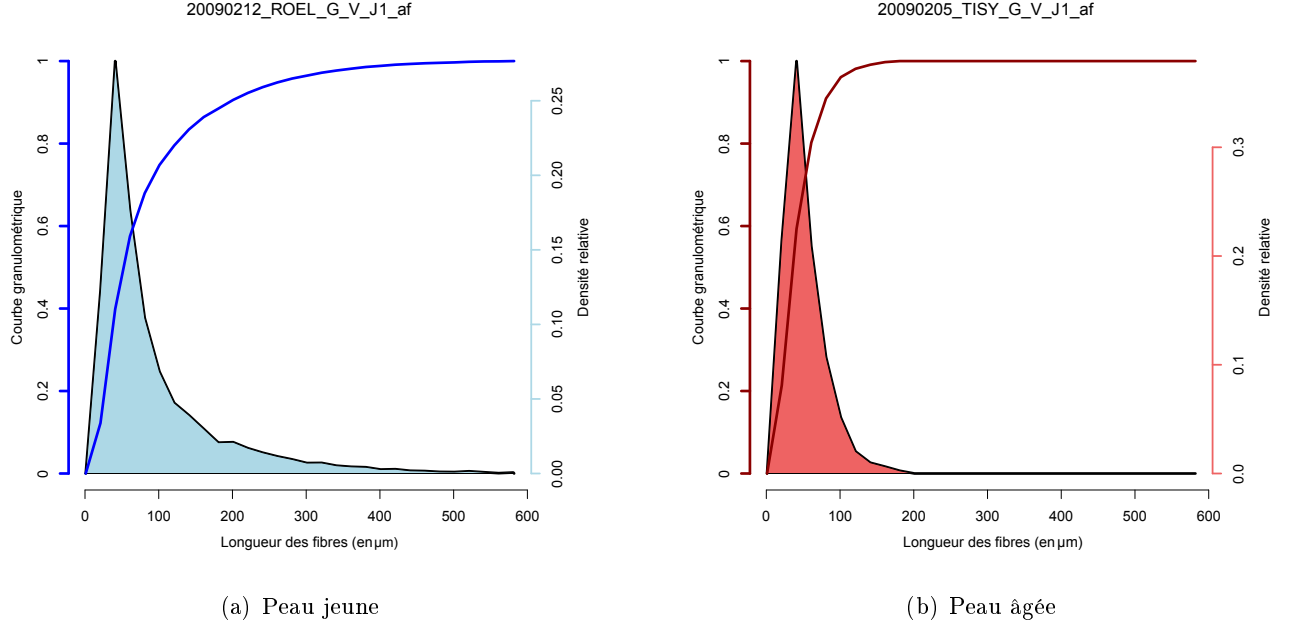


FIGURE 8.7 – Analyse de la distribution de tailles d’une peau jeune (a) et d’une peau âgée (b) avec des ouvertures parcimonieuses par chemins 3-D. Les courbes rouges et bleus foncées correspondent aux courbes granulométriques normalisées entre 0 et 1.

consiste à généraliser la sélection des chemins. Le graphe est alors adapté à des images 3-D (voir la figure 8.6 pour un graphe orienté suivant l’axe  $z$ ) et chaque voxel appartenant aux bords de l’image est un point de départ d’un chemin.

Le filtrage des chemins reste identique à celui des images 2-D en calculant des ouvertures linéaires le long des chemins avec l’algorithme développé dans le chapitre 2).

### Courbes granulométriques avec des PPO

Pour analyser la distribution de tailles des structures fibreuses du derme, on calcule des granulométries avec des ouvertures parcimonieuses par chemins 3-D. On applique une *top hat* sur l’image rehaussée, suivi par la suppression des valeurs inférieures à un niveau de gris égal à 18.

Pour une population d’objets  $X$  et une mesure de longueur  $length(x)$ ,  $x \in X$ , on définit la courbe granulométrique ou le *pattern spectrum cumulé* comme la probabilité que la longueur soit inférieure à  $L$  :  $G(L) = P(length \leq L)$ . Le *pattern spectrum* ( $ps$ ) correspond à la différentielle de  $G$ , (défini par Maragos [131], voir l’équation 2.7). Avec des ouvertures parcimonieuses par chemins, le pattern spectrum est calculé avec une mesure de volume et en utilisant le même ensemble de chemins sans procéder à une étape de reconstruction.

La figure 8.7 montre les *pattern spectra* et les courbes granulométriques d’une peau jeune et âgée pour deux patients ayant des structures fibreuses extrêmes (présence (resp. absence) de fibres longues pour la peau jeune (resp. âgée)).

Dans le cas discret, on a  $ps : \mathbb{N}^+ \rightarrow \mathbb{R}^+$ , la distribution de longueur des fibres. À partir de ce tableau, deux descripteurs scalaires sont définis pour décrire les distributions. Le premier paramètre,

appelé  $LengthPPO(mod, roi)$ , calcule la longueur moyenne des fibres :

$$LengthPPO = \frac{1}{\sum_{l=1}^{lMax} ps(l)} \sum_{l=1}^{lMax} l \times ps(l), \quad (8.10)$$

avec  $lMax$ , leur longueur maximale. Le second paramètre se focalise sur les longues fibres en calculant le pourcentage des fibres ayant une longueur supérieure à  $200 \mu m$  :

$$PourcLongPPO = \frac{\sum_{l=200}^{lMax} ps(l)}{\sum_{l=1}^{lMax} ps(l)} \times 100. \quad (8.11)$$

Ces deux descripteurs seront calculés sur toutes les régions d'intérêt du derme. Pour les deux distributions associées à la figure 8.7, les valeurs des descripteurs sont respectivement égales à  $LengthPPO = 92,7 \mu m$ ,  $PourcLongPPO = 11,5\%$  et à  $LengthPPO = 52,0 \mu m$ ,  $PourcLongPPO = 0\%$  pour les sujets jeune et âgé sur le derme total.

Les granulométries basées sur les ouvertures parcimonieuses par chemins sont très différentes des granulométries classiques avec des éléments structurants ayant une forme de sphère, qui permettraient de mesurer la largeur moyenne des fibres.

### 8.3.6 Organisation spatiale des fibres

Après avoir quantifié l'orientation, la forme ou la distribution de tailles des fibres, on s'intéresse à leur organisation spatiale. Ainsi, on squelettise les fibres et on analyse l'ensemble des branches du squelette.

#### Algorithme de squelettisation

On utilise un algorithme d'amincissement homotopique 3-D, fonctionnant avec des images en niveaux de gris [54]. Le squelette obtenu est donc topologiquement correct et suit le centre des fibres. En effet, on applique ce filtre sur des fibres ayant un profil gaussien, puisqu'elles ont été rehaussées par le filtre défini dans la section 8.2.2.

Une fois le squelette obtenu, on supprime les barbules de faible longueur (typiquement inférieure à la largeur moyenne des fibres) puis on supprime tous les points triples du squelette. On obtient un ensemble de branches noté  $\{B_i\}$ . Cette dernière étape est indispensable pour quantifier l'organisation spatiale des fibres, puisque nous n'avons plus une représentation de la longueur du réseau de fibres mais de leur agencement spatial les unes par rapport aux autres.

#### Extraction de descripteurs

À partir de l'ensemble des branches de squelette, on calcule plusieurs statistiques pour caractériser l'organisation des fibres : le nombre de branches (équation 8.12), la longueur moyenne des branches (équation 8.13) et le nombre de branches longues (équation 8.14) :

$$SkelNbBr = Card(\{B_i\}), \quad (8.12)$$

$$SkelLongMoyBr = \frac{\sum_{b_i \in \{B_i\}} length(b_i)}{Card(\{B_i\})}, \quad (8.13)$$

$$SkelNbBrLong = Card(b_i \in \{B_i\} | length(b_i) > 15 \mu m). \quad (8.14)$$

Les descripteurs définis dans cette section sont répartis en 4 catégories : les descripteurs caractérisant (i) l'orientation, (ii) la forme, (iii) la distribution de tailles et (iv) l'organisation des fibres. Cet ensemble de descripteurs sera utilisé pour améliorer la compréhension de l'évolution du réseau de fibres avec le temps.

## 8.4 Analyse statistique

Parmi l'ensemble des descripteurs obtenus, nous cherchons à extraire ceux qui évoluent au cours du temps. Des méthodes statistiques existent afin de définir ce sous ensemble de descripteurs dit *significatifs*. Néanmoins, ces méthodes partent toutes de l'hypothèse que les descripteurs ont une distribution normale (gaussienne). On propose donc de le vérifier dans la section suivante.

### 8.4.1 Test de normalité

En statistique, tester la normalité d'une variable revient à vérifier si les données réelles suivent une distribution gaussienne. Ces tests sont divisés en deux groupes : les méthodes graphiques (histogrammes, les boîtes à moustaches, *QQ plot* [251]) et les approches probabilistes (test de Cramer-Von Mises [4], Kolmogorov-Smirnov [122], Jarque-Bera [97] et beaucoup d'autres).

On s'intéresse à cette dernière catégorie. Pour visualiser l'influence du nombre d'échantillons sur les résultats ainsi que pour comprendre le fonctionnement de tels tests, nous simulons les données d'entrée suivant deux distributions : une loi normale et une loi uniforme. Les tests de normalité sont des tests d'hypothèse [62, 166] basés sur l'hypothèse nulle,  $H_0$  et sur l'hypothèse alternative  $H_a$  :

#### DÉFINITION 8.1

*Hypothèse nulle, notée  $H_0$ , correspond au point de vue de base à la position par défaut concernant un phénomène donné.*

Dans la plupart des tests d'hypothèse,  $H_0$  est formulée afin d'être rejetée. Cependant, ce n'est pas le cas avec les tests de normalité où l'on souhaite au contraire, que l'hypothèse nulle ne soit pas rejetée lorsque nos données d'entrée suivent une distribution gaussienne :

$$\begin{cases} H_0 : \text{les données suivent une loi normale,} \\ H_a : \text{les données ne suivent pas une loi normale.} \end{cases} \quad (8.15)$$

On utilise le test de Jarque et Bera [97] :  $JB = \frac{n}{6} \left( S^2 + \frac{(K-3)^2}{4} \right)$ , avec  $S$  le coefficient d'asymétrie (moment d'ordre 3) et  $K$  le kurtosis (moment d'ordre 4). Formellement, cette statistique ne teste pas si la distribution suit une loi normale mais si le coefficient d'asymétrie et le kurtosis des données sont les mêmes que ceux d'une loi normale. Les hypothèses  $H_0$  et  $H_a$  deviennent :

$$\begin{cases} H_0 : S = 0 \text{ et } K = 3, \\ H_a : S \neq 0 \text{ ou } K \neq 3. \end{cases} \quad (8.16)$$

Voici le procédé mis en place : on fixe un nombre d'échantillons,  $n$ , puis on tire aléatoirement  $n$  échantillons suivant une loi normale  $\mathcal{N}(0,1)$ . On calcule la *p-value*<sup>1</sup> associée au test de Jarque et Bera<sup>2</sup>

1. On montrera dans la section suivante comment on calcule une *p-value* et comment l'interpréter

2. On utilise le package R *tseries* : <http://cran.r-project.org/web/packages/tseries/index.html>

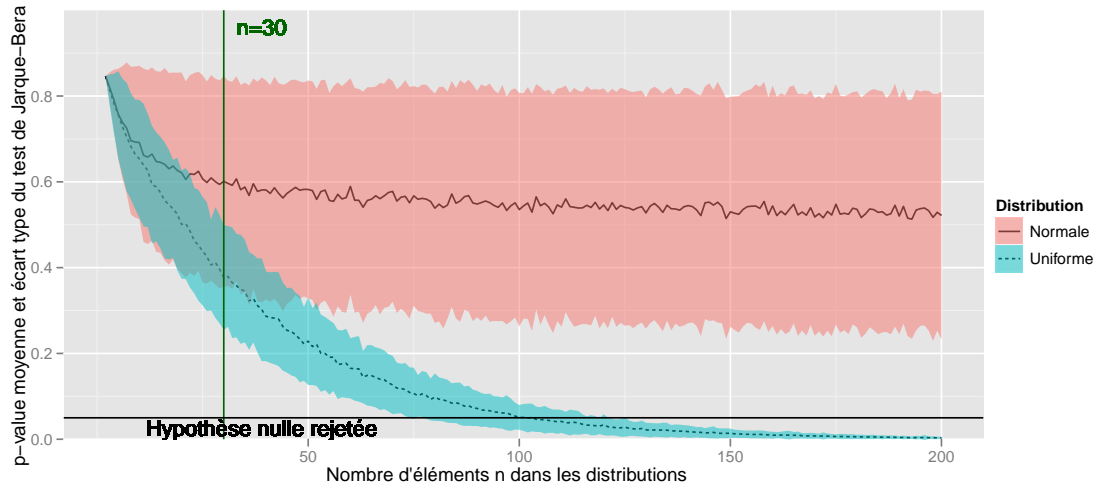


FIGURE 8.8 – Test de normalité de Jarque et Bera :  $p$ -value.

que l'on moyenne sur une centaine d'itérations. Enfin on applique le même processus pour différentes valeurs de  $n$ . Pour une loi normale, l'hypothèse nulle ne doit pas être rejetée ( $p\text{-value} > 0,05$ ).

On réalise le même procédé avec une loi uniforme. On s'attend donc à rejeter l'hypothèse nulle pour tous les  $n$ . La figure 8.8 regroupe les résultats de cette étude. La courbe et l'aire en bleu correspondent respectivement à la valeur moyenne des  $p$ -values et à la valeur moyenne plus ou moins l'écart type, pour une loi uniforme. En rouge, on a les résultats pour une loi normale.

On remarque deux phénomènes à partir de ce graphique. (i) l'hypothèse nulle n'est rejetée ( $p$ -values inférieures à 0,05) pour une loi uniforme que si le nombre d'échantillons est supérieur à 100. On ne peut donc pas faire de distinction entre une distribution normale et une distribution uniforme lorsque  $n = 30$  (le nombre d'échantillons par classe de notre base de données). (ii) l'écart type pour la loi normale est très important et certains tests rejettent l'hypothèse nulle à tort, même pour de grandes valeurs de  $n$ .

On note que les résultats sont similaires avec d'autres méthodes décrites dans la littérature. La multiplication des méthodes pour tester la normalité d'une distribution témoigne de la difficulté d'un tel test.  $n = 30$  est insuffisant pour rejeter l'hypothèse nulle et par conséquent, nous supposons que les distributions de nos variables suivent une loi normale. Les tests de significativité que l'on va introduire dans la section suivante nécessitent la normalité des distributions. Cependant, il a été montré que certains d'entre eux sont suffisamment robustes pour être utilisable même si les distributions s'écartent de la loi normale [179].

### 8.4.2 Test de significativité

Un descripteur est dit *significatif* au sens statistique du terme s'il est fortement improbable que l'on obtienne les mêmes résultats par chance. Fisher [63] en 1929, introduisit les notions de *test de significativité* qui ne sont rien d'autre que des *tests d'hypothèses*. Pour savoir si un paramètre de notre

étude a été modifié entre la population jeune et âgée, on met en place l'hypothèse nulle suivante :

$$\begin{cases} H_0 : \text{mean}(X_{Young}) = \text{mean}(X_{Old}) \\ H_a : \text{mean}(X_{Young}) \neq \text{mean}(X_{Old}), \end{cases} \quad (8.17)$$

avec  $X_{Young}$  et  $X_{Old}$  des vecteurs de dimension 30, résultats d'un descripteur pour les 30 échantillons jeunes et les 30 échantillons âgés.

Le test de significativité est illustré en choisissant un descripteur aléatoirement parmi ceux disponibles : *AnisXYZ*. La figure 8.9(a) présente sa valeur moyenne et son intervalle de confiance à 95%. Ce descripteur est-il significatif? Pour le savoir, on vérifie s'il rejette l'hypothèse nulle en utilisant le test-t de Welch [250].

Le test-t de Welch est une adaptation du test-t de Student [179], pour deux populations qui ont potentiellement des variances différentes :

$$ts = \frac{\text{mean}(X_{Young}) - \text{mean}(X_{Old})}{\sqrt{\frac{\text{std}(X_{Young})^2}{n_{Young}} + \frac{\text{std}(X_{Old})^2}{n_{Old}}}}, \quad (8.18)$$

avec les fonctions  $\text{mean}(\cdot)$  et  $\text{std}(\cdot)$  donnant respectivement la valeur moyenne et l'écart type des données et  $n_{Old} = n_{Young} = n/2$  le nombre d'échantillons par classe. Plus les distributions sont séparées et plus cette valeur augmente. Ce test a  $\nu$  degrés de liberté :

$$\nu = \frac{\left( \frac{\text{std}(X_{Young})^2}{n_{Young}} + \frac{\text{std}(X_{Old})^2}{n_{Old}} \right)^2}{\frac{\text{std}(X_{Old})^4}{n_{Old}^2(n_{Old}-1)} + \frac{\text{std}(X_{Young})^4}{n_{Young}^2(n_{Young}-1)}}. \quad (8.19)$$

Pour le descripteur sélectionné,  $t_s = 0,76$  et  $\nu = 41,84$ . La probabilité d'obtenir la valeur  $ts$  en considérant l'hypothèse nulle est calculée en prenant la distribution t qui a la densité de probabilité suivante :

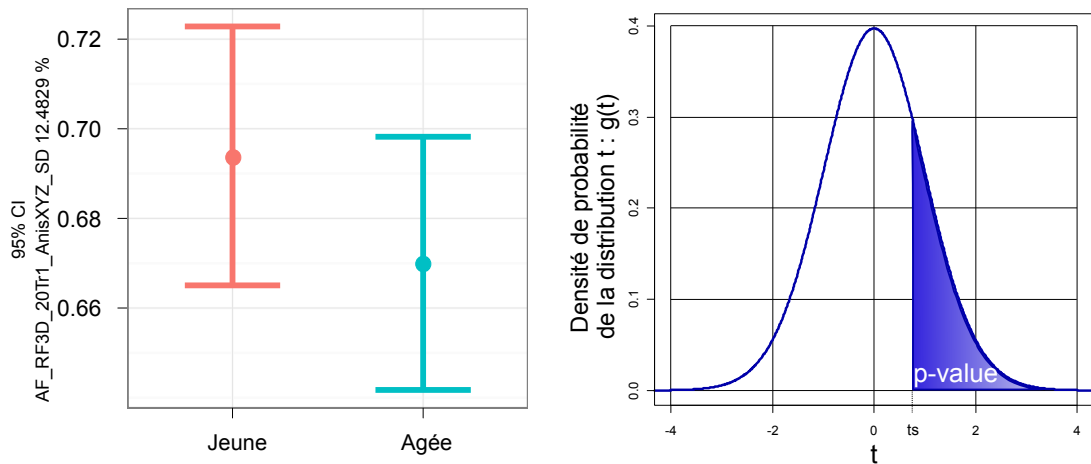
$$g(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})} \left( 1 + \frac{t^2}{\nu} \right)^{-\frac{\nu+1}{2}}, \quad (8.20)$$

avec  $\Gamma(\cdot)$ , la fonction gamma. La figure 8.9(b) présente la forme de la distribution t associée à ce descripteur.

Enfin, si nous supposons que l'hypothèse nulle est vraie, alors la *p-value* correspond à la probabilité d'obtenir par chance, un résultat au moins aussi extrême que celui que l'on observe. La *p-value* correspond donc à l'aire en bleu dans la figure 8.9(b) :

$$p\text{-value} = \int_{ts}^{\infty} g(t) dt. \quad (8.21)$$

Au final, on peut trier l'impact des descripteurs en fonction des *p-values* (plus la *p-value* est faible et plus les distributions sont séparées) et déterminer ceux qui sont pertinents pour notre étude en fixant un niveau à 5%. Pour ce descripteur la *p-value* est de 12,5% et par conséquent, l'hypothèse nulle n'est pas rejetée, cette variable n'est donc pas significative.



(a) Valeur moyenne et intervalle de confiance à 95% pour le descripteur *AnisXYZ* (b) Densité de probabilité de la distribution-t  $g(t)$  avec  $\nu = 41,84$ . L'intégrale  $\int_{ts}^{\infty} g(t)dt$  correspond à l'aire en bleu et est égale à la *p-value*

FIGURE 8.9 – Test de significativité avec un test-t de Welch sur le descripteur *AnisXYZ*. Avec un risque de première espèce de 5%, ce descripteur n'est pas significatif. Un risque de première espèce consiste à rejeter à tort l'hypothèse nulle alors qu'elle est vraie.

### 8.4.3 L'effet de taille

On complète cette analyse en calculant l'effet de taille (*effect size* en anglais) [164]. Cette mesure est calculée très simplement en prenant la différence entre les deux valeurs moyennes d'une variable et en divisant par l'écart type de l'une des deux populations. Cet effet mesure la force de la relation entre deux variables. L'effet de taille est un paramètre *descriptif* propre à une population et ne fait donc pas intervenir la taille de l'échantillon.

Afin d'être pertinent et significatif biologiquement entre deux groupes d'âge aussi extrêmes, l'estimation de l'importance de l'effet de taille de ces nouveaux paramètres a été établie en fonction de paramètres connus, tels l'épaisseur de l'épiderme, l'aplatissement de la JDE ou de l'élastose. L'effet de taille des paramètres précédemment cités étant respectivement de 0,91, 1,17 et de 1,37 (voir [49]), l'échelle suivante a été mise en place (tableau 8.2).

TABLEAU 8.2 – Importance de l'effet de taille fixé par les biologistes

	Intervalles pour l'effet de taille (en valeur absolue)
Très fort	$> 1,3$
Fort	$]0,8; 1,3]$
Modéré	$]0,5; 0,8]$
Faible	$]0,3; 0,5]$
Très faible	$]0; 0,3]$
Pas d'effet	0

## 8.5 Interprétation des résultats

Le vieillissement de la peau est un phénomène complexe cumulant plusieurs manifestations aux niveaux des cellules et des fibres. Cette transformation se manifeste notamment par une perte d'élasticité de la peau. Ainsi, comprendre comment les fibres évoluent avec le temps est important et on apporte, dans cette section, des éléments de réponses grâce aux descripteurs définis dans la section 8.3 de ce chapitre.

### 8.5.1 Orientation du réseau de fibres

Les mesures d'anisotropie sur le derme total (voir les figures 8.10(a) et 8.10(b)) révèlent une information intéressante : pour le canal SHG (fibres de collagène), le facteur d'anisotropie est plus important lorsque l'on prend comme référentiel la jonction dermo-épidermique par rapport au référentiel xyz. Cela signifie que les fibres de collagène sont orientées suivant une direction principale qui est fortement corrélée à la forme de la JDE.

Le tableau 8.3 donne les *p-values* ainsi que les effets de taille pour l'angle  $\Phi$  par rapport à la JDE. Le descripteur *PhiJDE* pour le canal SHG est significatif et, par conséquent, évoluent avec le temps. Pour des peaux âgées, les valeurs moyennes des angles passent de  $24^\circ$  à  $27^\circ$  pour le canal SHG. Cependant, pour des mesures d'angles, un effet de taille de 0,75 est considéré, par les biologistes, comme une évolution très modérée.

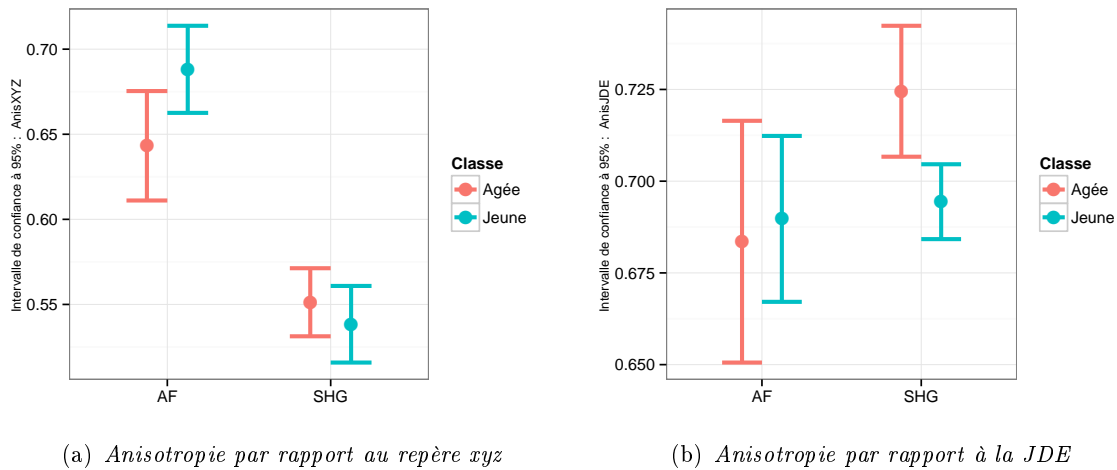


FIGURE 8.10 – Intervalles de confiance à 95% des descripteurs représentant l'anisotropie des fibres par rapport au repère xyz et par rapport à la JDE.



TABLEAU 8.3 – Test de significativité sur les descripteurs mesurant l'anisotropie des fibres et les angles  $\phi$  par rapport à la JDE. Ces descripteurs sont classés par  $p$ -value croissante.

Descripteurs	Effet de taille	$ts$	$\nu$	$p$ -value (%)
$AnisJDE(SHG,DT)$	-0,75	-2,87	46,13	0,3112
$PhiJDE(SHG,DT)$	0,71	2,7	55,14	0,4574
$AnisXYZ(AF,DT)$	0,56	2,14	55,22	1,833
$PhiJDE(AF,DT)$	0,3	1,15	56,63	12,6549
$AnisXYZ(SHG,DT)$	-0,22	-0,84	57,21	20,202
$AnisJDE(AF,DT)$	0,08	0,3	51,36	38,0896

### 8.5.2 Morphologie du réseau de fibres

Les mesures géodésiques permettent de caractériser la longueur, l'élongation ou la tortuosité moyenne des réseaux de fibres. En prenant le derme total comme région d'intérêt, ces trois descripteurs ne sont pas significatifs. Cela signifie qu'il n'y aurait donc pas de modifications notables de la morphologie du réseau de fibres entre une peau jeune et âgée, en moyenne dans le derme. On a donc entrepris de réaliser une étude plus précise en regardant l'évolution de ces paramètres dans les régions d'intérêt localisées sous la JDE.

La figure 8.11(b et c) présente l'intervalle de confiance à 95% de la longueur géodésique dans les tranches de  $20\ \mu m$ , numéros 1 et 2, sous la JDE. On note que si la longueur moyenne des structures de l'image 3-D a un effet de taille faible dans le derme total, cet effet augmente fortement dans les tranches sous la JDE pour le canal AF. Dans la tranche la plus proche de la JDE ( $Tr1$ ), la  $p$ -value est très faible ( $< 1e^{-5}$ ) et on constate que la longueur moyenne du réseau fibreux diminue avec l'âge. C'est un phénomène déjà constaté pour les fibres d'élastines (canal AF) [61]. Dans la littérature, aucune information n'est disponible sur les fibres de collagène (SHG), et pour ces fibres, la longueur ne devient pas un paramètre significatif : la longueur des fibres de collagènes n'évoluerait que peu avec le temps (effet de taille de 0,28 dans la tranche 2). De plus, on remarque que dans la tranche la plus éloignée de la JDE, les longueurs des fibres est plus importante pour des peaux âgées. Enfin, on note que la longueur géodésique moyenne du réseau de fibres de collagène est plus importante que celle des fibres élastines. Le tableau 8.4 regroupe les résultats de ces descripteurs.

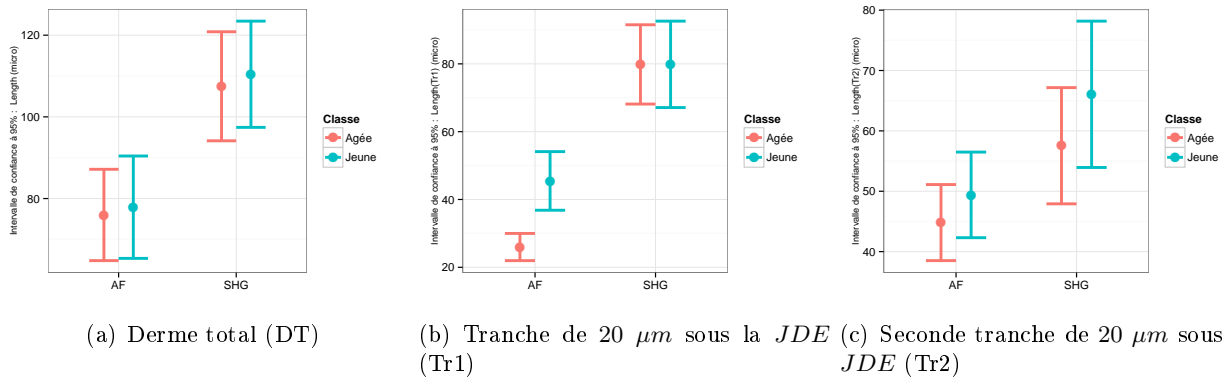


FIGURE 8.11 – Intervalle de confiance à 95% de trois descripteurs basés sur la longueur géodésique dans le derme total et dans les tranches sous la JDE.

TABLEAU 8.4 – Test de significativité des descripteurs basés sur la longueur géodésique, classés par  $p$ -value croissante.

Descripteurs	Effet de taille	$ts$	$\nu$	$p$ -value (%)
$Length(AF,Tr3)$	-1,57	-6	35,24	1e-5
$Length(AF,Tr1)$	1,05	4,01	40,93	0,0129
$Length(SHG,Tr3)$	-0,73	-2,77	56,54	0,3804
$Length(SHG,Tr2)$	0,28	1,08	55,17	14,2772
$Length(AF,Tr2)$	0,25	0,95	57,21	17,3492
$Length(SHG,DT)$	0,08	0,31	57,96	37,8968
$Length(AF,DT)$	0,06	0,22	57,26	41,2408
$Length(SHG,Tr1)$	0	0	57,57	49,9027

### 8.5.3 Distribution de taille du réseau de fibres

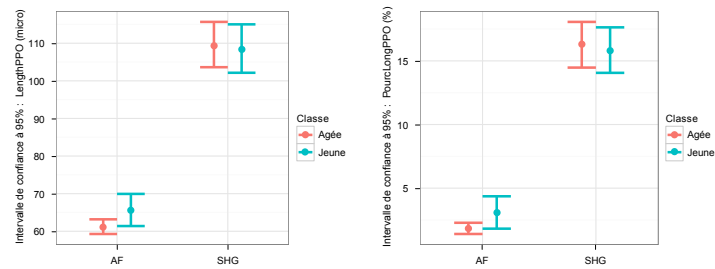
Les descripteurs basés sur les ouvertures parcimonieuses par chemins en 3-D permettent de caractériser d'une manière différente le réseau de fibres du derme. En effet, le graphe utilisé restreint leur tortuosité et ces opérateurs analysent l'image en niveau de gris. Pour ces deux mesures (voir figure 8.12), les résultats sont en accord avec les mesures de longueur avec le diamètre géodésique : en moyenne, les fibres de collagène sont plus longues que les fibres d'élastine (110  $\mu m$  pour les fibres de collagène contre 65  $\mu m$  pour les fibres d'élastine). On note cependant que les valeurs numériques sont très différentes, ce qui est normal. De plus, on remarque que la proportion de fibres longues est plus importante pour le canal SHG (collagène). De ces mesures, on peut en déduire que le réseau de fibres de collagène est plus dense que le réseau de fibres d'élastine.

Pour le canal AF, des différences notables apparaissent entre le réseau de fibres d'une peau jeune et celui d'une peau âgée. Les fibres d'élastine ont tendance à se raccourcir avec l'âge. Ce phénomène est amplifié plus on restreint la région d'intérêt aux tranches proches de la JDE,  $Tr1$  et  $Tr2$  (voir la figure 8.13). Enfin, dans la dernière tranche du derme ( $Tr3$ ), on observe le phénomène inverse, ce qui dénote une réorganisation complète du réseau de fibres. Le tableau 8.5 résume les  $p$ -values des différents descripteurs associés aux ouvertures parcimonieuses par chemins.

### 8.5.4 Organisation du réseau de fibres

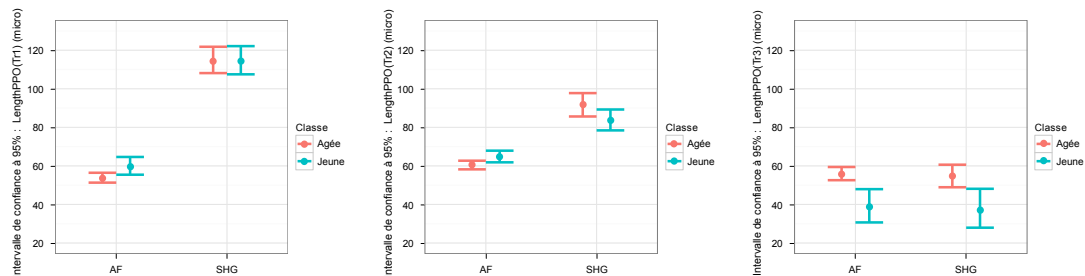
Les statistiques basées sur les branches du squelette du réseau de fibres fournissent des informations sur leur organisation spatiale. Des phénomènes intéressants apparaissent sur le derme total aussi bien pour les fibres de collagène que pour les fibres d'élastine (l'effet de taille, illustré dans le tableau 8.6, montre que ce phénomène est sensiblement plus marqué sur le canal SHG) : la longueur moyenne des branches du squelette, ainsi que le nombre de longues branches sont plus importants pour des peaux âgées (figure 8.14). L'ensemble des branches du squelette traduit l'imbrication des fibres et on peut émettre l'hypothèse que le réseau de fibres est plus lâche et moins imbriqué pour des peaux âgées. Ceci est une observation nouvelle qui permet de mieux comprendre l'évolution des fibres avec le temps.

## 8. PROJET L'ORÉAL



(a) Longueur moyenne du réseau (b) Pourcentage des structures de fibres sur le derme total (DT) supérieures à  $100\ \mu m$  sur le derme total

FIGURE 8.12 – Intervalle de confiance à 95% des deux descripteurs basés sur la distribution de longueur du réseau de fibres.



(a) Longueur moyenne dans la tranche 1 sous la JDE (Tr1) (b) Longueur moyenne dans la tranche 2 sous la JDE (Tr2) (c) Longueur moyenne dans la tranche 3 sous la JDE (Tr3)

FIGURE 8.13 – Intervalle de confiance à 95% des descripteurs basés sur la distribution de tailles du réseau de fibres dans les tranches sous la JDE.

TABEAU 8.5 – Récapitulatif des tests de significativité.

Descripteurs	Effet de taille	$ts$	$\nu$	$p$ -value (%)
$LengthPPO(AF,Tr3)$	-1,1	-3,52	26,79	0,0803
$PourcLongPPO(AF,Tr3)$	-1,02	-3,27	21,05	0,1907
$LengthPPO(SHG,Tr3)$	-0,88	-2,81	32,89	0,4167
$LengthPPO(AF,Tr2)$	0,6	2,3	53,47	1,267
$LengthPPO(AF,Tr1)$	0,6	2,29	45,33	1,3357
$PourcLongPPO(AF,Tr1)$	0,55	2,08	36,91	2,2266
$PourcLongPPO(SHG,Tr2)$	-0,52	-1,99	56,87	2,5744
$LengthPPO(SHG,Tr2)$	-0,5	-1,89	57,29	3,1641
$LengthPPO(AF,DT)$	0,49	1,85	40,55	3,5803
$PourcLongPPO(AF,DT)$	0,48	1,82	35,85	3,8707
$PourcLongPPO(AF,Tr2)$	0,47	1,78	44,43	4,1194
$PourcLongPPO(SHG,Tr3)$	-0,38	-1,22	40,74	11,4544
$PourcLongPPO(SHG,DT)$	-0,09	-0,33	58	37,313
$PourcLongPPO(SHG,Tr1)$	-0,06	-0,24	57,66	40,4554
$LengthPPO(SHG,DT)$	-0,06	-0,24	57,74	40,6947
$LengthPPO(SHG,Tr1)$	-0,01	-0,03	57,73	48,8172

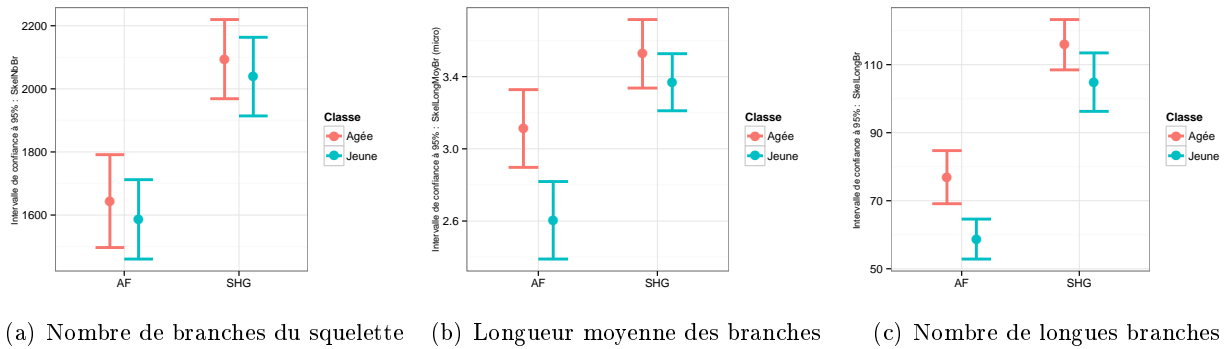


FIGURE 8.14 – Intervalle de confiance à 95% de trois descripteurs basés sur les branches du squelette des fibres sur le derme total.

TABLEAU 8.6 – Test de significativité sur les descripteurs mesurant l’organisation spatiale des fibres, classés par  $p$ -value croissante.

Descripteurs	Effet de taille	$ts$	$\nu$	$p$ -value (%)
$SkelNbBrLong(AF,DT)$	-0.96	-3.65	53.87	0.03
$SkelLongMoyBr(AF,DT)$	-0.86	-3.28	58	0.0891
$SkelNbBrLong(SHG,DT)$	-0.5	-1.9	56.74	3.1166
$SkelLongMoyBr(SHG,DT)$	-0.33	-1.25	56.17	10.8383
$SkelNbBr(SHG,DT)$	-0.16	-0.61	58	27.0926
$SkelNbBr(AF,DT)$	-0.15	-0.59	56.63	28.0122

## 8.6 Bilan et perspectives

Ce chapitre applicatif présente le travail réalisé dans le cadre du projet L’Oréal. Des descripteurs morphologiques, tous 3-D, ont été introduits dans cette thèse pour caractériser la forme, l’orientation, la distribution de tailles et l’organisation des réseaux de fibres présents dans la partie supérieure du derme de la peau humaine.

Une application au vieillissement de la peau est proposée pour comprendre l’évolution des structures fibreuses avec le temps. Aussi, une base de données d’images est disponible, composée pour moitié d’images provenant de peaux jeunes et pour autre moitié, de peaux âgées. Les descripteurs sont calculés sur chaque modalité (AF ou SHG) et dans plusieurs régions d’intérêt : le derme tout d’abord, puis des régions d’intérêt plus fines, localisées sous la JDE, pour affiner l’analyse dans les cas où les mesures sur le derme total ne montrent pas de modifications majeures.

Une analyse statistique complète cette étude. Plusieurs méthodes de la littérature sont utilisées pour tester la normalité d’une variable. Cependant, aucune d’entre elles ne permet de tirer des conclusions sur la normalité d’une distribution lorsque le nombre d’échantillons est très faible. Il serait donc intéressant de refaire cette étude pour un nombre d’échantillons plus important, proche de 100. De plus, ceci, permettra d’avoir une plus grande précision sur les mesures de *l’effet de taille* et des  $p$ -values. Ces deux outils statistiques mis en place permettent de connaître les paramètres qui évoluent entre une peau jeune et une peau âgée et dans quelle mesure. Cette analyse statistique permet de faciliter l’interprétation des résultats en classant ces paramètres par  $p$ -values croissantes.

## 8. PROJET L'ORÉAL

---

Grâce aux descripteurs définis dans ce chapitre et à l'analyse statistique réalisée, une interprétation des résultats est proposée. On observe que l'orientation principale des fibres est fortement corrélée à la forme de la jonction dermo-épidermique pour le canal SHG. De plus, la morphologie de ce réseau est modifiée dans les tranches sous cette jonction. La longueur du réseau de fibres de collagène et d'élastine diminue avec l'âge. Enfin, il semble que le réseau de fibres devienne plus lâche, moins imbriqué, pour des peaux âgées.

En microscopie multiphoton, il s'agit de la première étude réalisée en trois dimensions dans sa totalité pour caractériser les réseaux de fibres du derme de la peau humaine.

De futures études prendront en compte d'autres bases de données, notamment la base de données relative aux facteurs environnementaux et en particulier le soleil. Des images sont acquises sur la partie dorsale de l'avant bras, une zone plus exposée aux rayons UV que la partie ventrale. En comparant l'évolution des paramètres définis dans ce chapitre, nous pouvons observer l'impact du soleil sur les structures fibreuses de la peau.

Associer l'imagerie multiphoton à des outils de traitement d'images tridimensionnelles permet de visualiser et de quantifier les structures de la peau. C'est donc une aide majeure pour évaluer l'effet des traitements dermatologiques ou les produits cosmétiques.

---

### Projet Colas : Contrôle non destructif des arrachements des routes par le traitement d'images et analyse statistique

---

*« Clearly, I reject the view that there is one way that is right for everyone and for every problem. »*

Bjarne Stroustrup

Dans le cadre des contrôles non destructifs, étudier l'évolution des routes avec le temps fournit des informations essentielles sur le vieillissement des voies de circulation. Leur contrôle est donc indispensable pour concevoir des routes plus résistantes ou pour rénover les tronçons trop abimés. Pour réaliser ce contrôle d'une manière non invasive et automatique, une caméra mobile est associée à des outils de traitements d'images pour extraire les défauts de la chaussée.

Dans ce contexte, l'étude présentée dans ce chapitre propose une méthode originale pour localiser les défauts de surface des routes et plus précisément, les zones du bitume qui sont arrachées. Une étape de quantification permet de connaître la surface représentée par ces zones et donc le taux de dégradation. Ce problème est assimilé à de la caractérisation de textures des images. On va donc extraire des imagerie représentatives des textures à analyser (des patchs de 10 cm par 10 cm), on calculera des descripteurs morphologiques puis, on réalisera deux analyses statistiques, une non supervisée pour obtenir la vérité terrain et l'autre supervisée afin de réaliser un classement des zones suspectes dans l'image.

De la même manière que dans le projet TOCATA (chapitre 7) pour l'extraction de défauts de surface de pièces métalliques, la stratégie globale inclue une étape de recherche de descripteurs pertinents puis une étape d'apprentissage statistique supervisée pour adapter le traitement aux différentes routes existantes.

Un livrable logiciel doit être fourni aux demandeurs de ce projet et on propose un logiciel simple d'utilisation, fonctionnel et intuitif permettant de choisir les descripteurs voulus et de facilement créer la vérité terrain en vue de réaliser une analyse statistique supervisée.

Ce projet est réalisé en partenariat avec le groupe Colas<sup>1</sup>, qui est une entreprise française leader mondial de la construction et de l'entretien de routes.

---

1. Colas : <http://www.colas.com>

Mots clés : Contrôle non destructif, routes, analyse de la texture, traitement d'images, analyse statistique, ACP, régression logistique, classement par patches.

### 9.1 État de l'art

La recherche de défauts de surface des routes en utilisant à un système optique est une problématique cruciale pour le contrôle automatique des voies de circulation. Les premières caméras embarquées apparaissent en 1991 avec les systèmes *ADDA*<sup>1</sup>, *ACM*<sup>2</sup> ou *SIRANO*<sup>3</sup>. Cependant, ils n'utilisent pas d'outils de traitement d'images et sont surpassés par les systèmes *REAL*<sup>4</sup> en 1992, *ARAN*<sup>5</sup> en 1993, *ADVantage* en 2003, *RoadCrack*<sup>6</sup> [60] en 2003, *AMAC*<sup>7</sup> en 2004 ou le système *LRIS*<sup>8</sup> de l'Institut National de l'Optique. Chaque système optique a des caractéristiques propres et le lecteur est invité à lire [216, 160] pour un comparatif détaillé.

D'un point de vue traitement d'images, de nombreuses méthodes ont été mises au point pour la détection de défauts de surface de type fissures dans l'image : des méthodes basées sur l'histogramme, les ondelettes, les réseaux de neurones, le filtrage, la morphologie mathématique... Chambon et Moliard [33] proposent un état de l'art complet de ces méthodes tout en les comparant les unes aux autres. Pour l'extraction des défauts de type fissures des routes, toutes les méthodes d'analyse d'images développées dans ce document peuvent être utilisées. En effet, les fissures sur les routes et les fissures des pièces métalliques sont relativement similaires. De plus, utiliser la stratégie mise en œuvre dans le projet TOCATA (chapitre 7) peut permettre d'apprendre la signature des défauts et des images.

Dans cette étude, les défauts à rechercher ne sont pas des fissures mais des irrégularités de surface traduites par des modifications de texture. Par conséquent, hormis l'analyse statistique AdaCOS qui pourra être adaptée pour cette application, les nouveaux opérateurs développés dans la partie 1 de ce document de thèse ne seront pas utilisés pour ce projet. À notre connaissance, c'est la première fois qu'une méthode automatique est utilisée pour extraire les défauts de surface de type arrachement de la chaussée.

### 9.2 Présentation du projet Colas

Ce projet propose un contrôle non destructif et automatique des voies de circulation afin de connaître précisément l'état de surface des routes. Une phase de quantification est nécessaire afin de créer des indicateurs permettant de remplacer les tronçons de route trop endommagés. Ce contrôle nécessite un système optique que l'on présentera dans la section suivante. Puis, nous visualiserons des images des défauts surfaciques à détecter. Enfin, nous présenterons la stratégie générale mise en place pour répondre à cette problématique. Dans cette étude, nous cherchons à analyser les différentes textures des images pour localiser différents défauts de surface : les reprises, les nids de poule, les pelades, les arrachements ou le ressutage.

- 
1. ADDA : Automated Distress Data Acquisition
  2. ACM : Automated Crack Monitor
  3. SIRANO : Système d'Inspection des Routes et Autoroutes par Analyses Numériques et Optiques
  4. REAL : Road Excellent Automatic Logging (<http://www.pasco.co.jp/eng/products/real/>)
  5. ARAN : Automatic Road Analyser ([http://www.maine.gov/mdot/systems\\_management/aran.php](http://www.maine.gov/mdot/systems_management/aran.php))
  6. RoadCrack : <http://www.csiro.au/Outcomes/ICT-and-Services/People-and-businesses/RoadCrack.aspx>
  7. AMAC : Appareil Multifonction d'Auscultation des Chaussées (<http://www.vectra.fr/sites/fr/materiels-et-solutions/auscultations-des-chaussees/amac.html>)
  8. LRIS : Laser Road Imaging System (<http://www.ino.ca/en-ca/achievements/description/project-p/laser-road-imaging.html>)

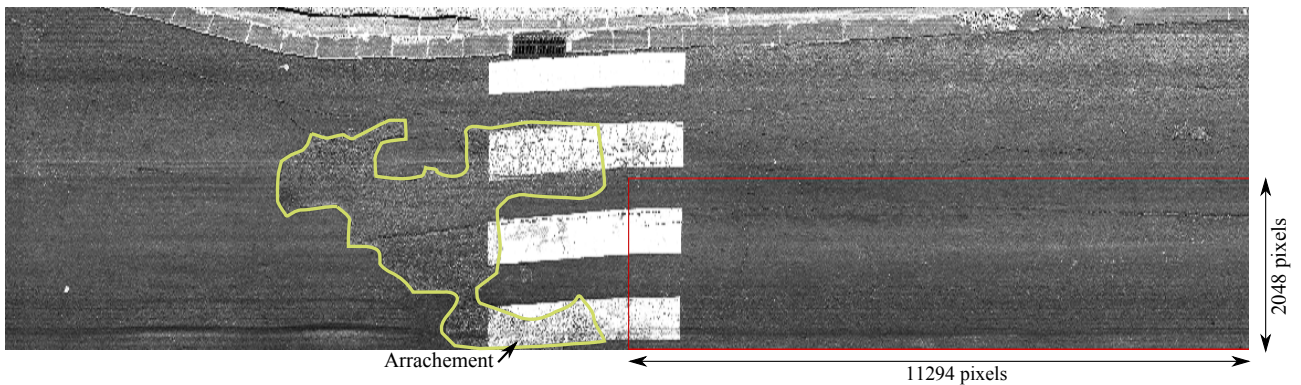


FIGURE 9.2 – Acquisition d’une série d’images de route par le système de vision de Colas dénommé : *Road Eagle Colas*. Le défaut de surface de type arrachement est entouré de jaune et cette section est composée de 4 images juxtaposées dont les dimensions sont de 11294 par 2048 pixels par image, correspondant au rectangle rouge.

### 9.2.1 Système de vision : acquisition des images



FIGURE 9.1 – Système de vision

Le système de vision provient de l’Institut National d’Optique (INO<sup>1</sup>), qui propose d’inspecter les routes à une résolution de 1 mm par pixel. Deux capteurs linéaires sont montés sur un véhicule adapté pour faire l’acquisition des images. La figure en tête de paragraphe présente la disposition des caméras et de l’éclairage laser qui permet de s’affranchir des ombres et d’uniformiser l’éclairage des lignes. Le camion roulant à vitesse variable, une synchronisation de la fréquence d’acquisition des caméras est nécessaire pour obtenir deux images 2-D, une pour chaque capteur. Chaque image est constituée de 2048 par 11294 pixels et la reconstitution de la route est aisée en plaçant côte à côte les images des deux caméras, puis bout à bout la succession des images. La figure 9.2 présente une portion de route acquise par ce système de vision.

### 9.2.2 Zones en arrachement

L’usure de la route se traduit par plusieurs phénomènes. On peut voir l’apparition de fissures ou de craquelures, néanmoins, cette étude ne porte pas sur la détection de ce type de défaut de surface. On peut voir aussi des arrachements de morceaux de route qui se traduisent par un changement de texture. La figure 9.2 présente un tel défaut, entouré de jaune sur cette image. On note que la texture saine est moins rugueuse que la texture de la route en arrachement mais ce phénomène ne se traduit pas toujours de la même manière et pour certains tronçons de routes, les arrachements sont moins rugueux que la texture saine. Il est donc très difficile de trouver des constantes dans les textures en arrachement puisque chaque route est différente en fonction du type d’enrobé ou d’enduit superficiel, des reprises ou du vieillissement. Par conséquent, on utilisera une stratégie générique qui s’adapte à chaque route grâce à un apprentissage supervisé. La stratégie utilisée est décrite dans la section suivante.

1. Institut National d’Optique : <http://www.ino.ca>



### 9.2.3 Stratégie générale

L'observation des images montre qu'il y a peu de constantes dans les textures à analyser. Aussi, leur caractérisation sera réalisée en utilisant une méthode générique couplée à des méthodes d'apprentissage statistique, pour obtenir un traitement spécialisé pour chaque type de défaut. La signature de la texture recherchée sera apprise et un modèle statistique généré afin de rechercher dans l'image les zones ayant cette même signature. Pour réaliser cet apprentissage statistique supervisé, il est nécessaire d'avoir la vérité terrain, élément qui doit être fourni manuellement. Pour réaliser cette tâche, l'utilisateur sera aidé d'une première segmentation obtenue avec des méthodes non supervisées. On résume ici les différentes étapes nécessaires pour la caractérisation des zones en arrachement :

1. Détection de la région d'intérêt : on réalise tout d'abord la détection automatique des lignes blanches puis on délimite toutes les régions de l'image qui ne correspondent pas à la chaussée : les caniveaux, les plaques d'égout, la terre, l'herbe...
2. Découpage de l'image : on découpe les images initiales en imagerie de taille 100 par 100 pixels. Avec la résolution utilisée, cela correspond à des patches de 10 cm par 10 cm sur la route. On note qu'il est obligatoire de travailler en pleine résolution pour ne pas détériorer les détails des textures. Un classement par patch nécessite la création de la vérité terrain, afin d'assigner une classe à chaque patch.
3. Calcul d'un ensemble de descripteurs sur chaque patch pour former un vecteur caractéristique, la signature du patch.
4. Détermination de la vérité terrain par un apprentissage statistique non supervisé. Suit une étape de correction manuelle pour affiner cette classification.
5. Génération d'un modèle statistique : des régressions logistiques multivariées sont utilisées pour construire un modèle statistique. Cet apprentissage supervisé permet de classer les patches des différentes images et ainsi de détecter les zones en arrachements.
6. Quantification : une fois la segmentation obtenue on peut donc calculer les caractéristiques des zones en arrachement : la surface totale et la surface localisée sur les bandes de roulement.

Dans un premier temps, nous présentons tous les éléments de cette étude. Enfin, nous discuterons du logiciel créé pour montrer la simplicité de mise en œuvre des différentes étapes et le degré d'automatisation du processus.

## 9.3 Détection de la région d'intérêt

Nous cherchons à construire un modèle de la texture de la route. Aussi, tous les éléments qui ne font pas partie de la chaussée ne doivent pas être pris en compte lors de la création de ce modèle. Le masquage de ces éléments est réalisé en deux étapes. La première étape consiste à supprimer automatiquement les lignes blanches des images, la seconde étape est manuelle et permet de masquer les éléments perturbateurs restant.

### 9.3.1 Détection automatique des lignes

Une succession de filtres morphologiques est utilisée pour filtrer le bruit et détecter les lignes blanches de l'image. Tous les filtres sont appliqués sur une image sous échantillonnée par 20, afin d'accélérer le traitement sans détériorer la détection :

- Tout d’abord, on filtre l’image par une ouverture morphologique par un carré de taille 3x3 pixels. Ceci supprimera les pixels blancs isolés.
- On applique une fermeture avec un élément structurant carré de taille 5 par 5 pixels pour combler les trous éventuels dans les lignes.
- Puis on calcule une ouverture par reconstruction sur l’image fermée. L’élément structurant à une forme carrée de 29 par 29 pixels pour supprimer toutes les lignes blanches. Ces structures supprimées se retrouvent dans les résidus : soustraction de l’image initiale avec l’image filtrée.
- La détection des lignes est combinée par un ET logique entre deux détections primaires : le seuillage des résidus et le seuillage des intensités.
- Enfin, on nettoie le résultat de la détection avec une petite ouverture.

Avec cette méthode, toutes les lignes blanches sont détectées, permettant de définir automatiquement une première région d’intérêt des images.

### 9.3.2 Masquage des autres éléments

Dans une image de la route, d’autres structures peuvent être présentes comme les plaques d’égouts ou les caniveaux. Ce sont des éléments que l’on ne souhaite pas détecter et qui doivent être masqués avant de réaliser un apprentissage statistique. Aussi, on propose une interface simple permettant de tracer des rectangles sur une image afin de masquer ces zones.

Lors de la division de l’image en patches, ceux qui ne seront pas entièrement dans la région d’intérêt ne seront pas pris en compte. L’image est alors découpée en patches de taille 100 par 100 pixels.

## 9.4 Descripteurs de texture

On met à disposition de l’utilisateur un nombre important de descripteurs que l’on va définir dans cette section. Chaque descripteur sera utilisé pour caractériser et introduire des signatures différentes entre les textures. On souhaite ainsi créer plusieurs frontières distinctes dans l’espace augmenté des descripteurs entre une texture saine et des textures en arrachement (hyperplans). On part de l’hypothèse suivante :

**Hypothèse :** *chaque patch n’est constitué que d’une seule et unique texture.*

En pratique, cette hypothèse n’est jamais respectée sur tous les patches puisque certains d’entre eux peuvent être à cheval sur deux textures différentes. Cependant, en partant de cette hypothèse, chaque patch n’appartiendra qu’à une seule classe rendant possible l’utilisation de méthodes statistiques.

Plusieurs techniques de caractérisation des textures sont utilisées et présentées dans cette section. Elles analysent l’image en pleine résolution puisque l’analyse des textures revient à caractériser les hautes fréquences de l’image. On découpe l’image en patches que l’on analysera un à un. Soit  $f$  une image  $D \rightarrow V$  avec  $D$  une grille carrée de taille 100 par 100 pixels et  $V$  l’ensemble des valeurs sur 8 bits.

### 9.4.1 Analyse de l’histogramme

L’analyse de l’histogramme des niveaux de gris d’un patch, fournit des informations sur les statistiques du premier ordre : la moyenne, l’écart type, la médiane, le coefficient d’aplatissement et le coefficient d’asymétrie. Ces **5 mesures** sont donc utilisées comme descripteurs pour caractériser la texture.

Ces descripteurs sont très faciles et rapides à calculer mais ils n'ont pas un pouvoir discriminant très important.

### 9.4.2 Local Binary Pattern

Les *local binary patterns* (LBP) sont des descripteurs locaux très populaire dans l'analyse des textures et la reconnaissance d'objet. Cette méthode, définie par Ojala *et al.* [171], compare les niveaux de gris du pixel central à ses pixels voisins pour former un motif binaire. Pour analyser une texture, il faut compter les répétitions de certains motifs qui apparaîtront fréquemment. La méthodologie que l'on a mise en place pour le projet Colas suit exactement la définition de Nguyen *et al.* [168] : des LBP uniformes et multi-échelles. [172].

#### Invariance par niveaux de gris

Une texture  $T$  est définie dans un voisinage local composé de 8 pixels (voisinage V8) comme la distribution jointe :

$$T = t(f_c, f_1, f_2, \dots, f_8), \quad (9.1)$$

avec  $f_c$  le niveau de gris du pixel central et  $f_i, \forall 1 \leq i \leq 8$  les niveaux de gris des pixels voisins. Cette texture peut être représentée sans perte d'information, comme la différence de niveaux de gris avec le pixel central :

$$T = t(f_c, f_1 - f_c, f_2 - f_c, \dots, f_8 - f_c). \quad (9.2)$$

On peut donc émettre l'hypothèse que les différences  $f_i - f_c$  sont indépendantes de  $f_c$  et factoriser :

$$T \approx t(f_c) t(f_1 - f_c, f_2 - f_c, \dots, f_8 - f_c) \approx t(f_1 - f_c, f_2 - f_c, \dots, f_8 - f_c). \quad (9.3)$$

En pratique, cette hypothèse d'indépendance n'est pas exacte mais permet d'obtenir un opérateur invariant par changement de niveaux de gris.  $t(f_c)$  décrit la luminance globale de la texture et n'est pas corrélée à cette dernière. On peut donc supprimer ce terme (voir équation 9.3).

Le signe des différences de niveaux de gris  $s(f_i - f_c)$  n'est pas affectée par des changements de la luminance moyenne, aussi, on considère uniquement le signe des différences :

$$T \approx t(s(f_1 - f_c), s(f_2 - f_c), \dots, s(f_8 - f_c)). \quad (9.4)$$

où

$$s(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon.} \end{cases} \quad (9.5)$$

En pondérant par un facteur  $2^i$  chaque signe  $s(f_i - f_c)$ , on obtient un code unique pour chaque motif :

$$LBP_8 = \sum_{i=1}^8 s(f_i - f_c) 2^i. \quad (9.6)$$

### Invariance par rotation

Avec 8 pixels, on obtient 256 motifs différents. Néanmoins, certains motifs sont identiques si on souhaite obtenir la propriété d'invariance par rotation. Avec une rotation de l'image originale, le pixel central sera toujours le même et seuls ses voisins auront subi cette rotation. Il est donc facile d'obtenir un code unique par motif invariant par rotation, en réalisant des rotations circulaires du code (fonction *ROR*) et en prenant la valeur minimale. On obtient  $LBP_8^{ri}$  :

$$LBP_8^{ri} = \min \{ROR(LBP_8, i) \mid i = 0, \dots, 7\}. \quad (9.7)$$

### LBP uniforme

On obtient donc 36 codes qui sont invariants par rotation. Néanmoins, certaines configurations n'ont pas la même fréquence et sont plus importantes que d'autres. Les motifs dits *uniformes* ( $LBP_8^{riu2}$ ) sont donc définis en prenant les 9 motifs qui possèdent au plus deux transitions 0–1 ou 1–0 dans leur code. Tous les autres motifs sont regroupés dans une dernière catégorie. On obtient donc 10 descripteurs pour caractériser une texture.

### LBP multi-échelles

Dans [168], une analyse multi-échelles est réalisée en sous échantillonnant l'image par 2 (tout en appliquant un filtrage moyen). On calcule donc les  $LBP_8^{riu2}$  pour chaque pixel de l'image sous échantillonnée puis nous concaténons les histogrammes obtenus. Dans cette étude, nous analysons trois échelles de mesures et nous obtenons au final **30 descripteurs**. Utiliser cette méthode permet de réduire les temps de calcul puisque il est inutile d'inspecter des voisinages plus grands que  $3 \times 3$ . De plus, le nombre de pixels à analyser est divisé par 4 à chaque échelle.

## 9.4.3 Matrice de co-occurrence et caractéristiques de Haralick

L'utilisation de la matrice de co-occurrence (ou matrice de dépendance spatiale) est une méthode largement utilisée pour caractériser une texture. Cette méthode permet de calculer les statistiques du second ordre par l'étude des relations spatiales entre des couples de pixels [85].

### Matrice de co-occurrence

On s'intéresse ici aux relations existantes entre les niveaux de gris d'une image pour un bipoint séparé par un vecteur  $\vec{d}$ . Il s'agit donc de mesurer la covariance d'un signal. Le résultat est ici présenté sous la forme d'une matrice  $M$  par  $M$  avec  $M$  le nombre de niveaux de gris de l'image. Ceci permet d'obtenir une représentation compacte de l'information de l'image. Formellement, cette matrice (notée  $C$ ) s'écrit ainsi :

$$C(i, j) = \text{Card} \left\{ r \in D, r + \vec{d} \in D \mid f(r) = i, f(r + \vec{d}) = j \right\}. \quad (9.8)$$

En pratique, les niveaux de gris d'une image sont souvent codés avec 8 bits donnant une matrice de co-occurrence, de taille  $256 \times 256$ , très sensible au changement d'intensité. En effet, une modification même minime des niveaux de gris d'une image, génèrera une matrice complètement différente. Pour augmenter la robustesse, on réduira l'information présente dans l'image en conservant uniquement 12 niveaux de gris. Un exemple de calcul de cette matrice est proposé dans la figure 9.3.

Texture :

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 2 & 2 \end{bmatrix}$$

$\Leftrightarrow$

Niveaux de gris $i \backslash j$	$\vec{d} = \rightarrow$			$\vec{d} = \nearrow$		
	0	1	2	0	1	2
0	1	3	1	1	3	0
1	3	2	1	3	1	0
2	0	0	1	0	0	1

FIGURE 9.3 – Illustration de la création de la matrice de co-occurrence avec deux vecteurs  $\vec{d}$  différents.

### Matrice de co-occurrence invariante par rotation

Chaque matrice de co-occurrences dépend fortement du vecteur  $\vec{d}$  et créer une matrice invariante par rotation revient à fusionner différentes matrices obtenues avec des vecteurs unitaires orientés dans les 8 directions principales de l'espace. Pour cette étude, le voisinage est donc fixé à  $3 \times 3$  pour la création de cette matrice.

L'analyse de la matrice de co-occurrence fournit 12 descripteurs de texture. Leur définition est donnée dans Haralick *et al.* [85].

### Matrice de co-occurrences multi-échelles

De la même manière que pour les LBP, une analyse multi-échelles est menée en sous-échantillonnant l'image initiale et en appliquant la même procédure sur cette image. Trois échelles sont analysées fournissant au total **36 descripteurs**.

#### 9.4.4 Grey level size zone matrix

La méthode *Grey level size zone matrix* (GLSZM) est une méthode développée par Thibault *et al.* [231, 230] pour caractériser la texture des noyaux de cellules. Cette méthode est une modification des matrices appelées *grey level run length matrix* [72] et des matrices de co-occurrence (voir la section précédente).

Une texture peut aussi être caractérisée par la taille de ses régions homogènes. C'est ce que mesure cet opérateur. La matrice associée à la mesure de la taille des régions homogènes est composée de  $M$  lignes (avec  $M$  le nombre de niveaux de gris<sup>1</sup>) et d'un nombre variable de colonnes (ce nombre correspond à la taille de la plus grande région homogène). La figure 9.4 présente un exemple d'une matrice GLSZ sur la même texture que dans la section 9.4.3.

On utilise le même ensemble de descripteurs que pour la caractérisation des matrices de co-occurrence, cette fois ci pour la caractérisation des matrices GLSZ. On obtient donc **12 descripteurs**, intrinsèquement invariants par rotation, pour décrire la texture d'une image.

#### 9.4.5 Auto-corrélation

On peut aller plus loin dans l'analyse de la texture par des bi-points. Le calcul de l'auto-corrélation d'une image (mesure de la corrélation d'une image par cette même image) est une généralisation de la matrice de co-occurrence et permet d'obtenir des informations sur les statistiques de second ordre

1. De la même manière que les matrices de co-occurrences, le nombre de niveaux de gris est fixé à 12 pour limiter l'impact du bruit

Texture :

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 2 & 2 \end{bmatrix}$$

$\Leftrightarrow$

Niveaux de gris	Taille des zones homogènes					
	1	2	3	4	5	6
0	0	1	0	1	0	0
1	1	0	0	0	0	1
2	0	0	1	0	0	0

FIGURE 9.4 – Illustration de la création de la matrice GLSZ

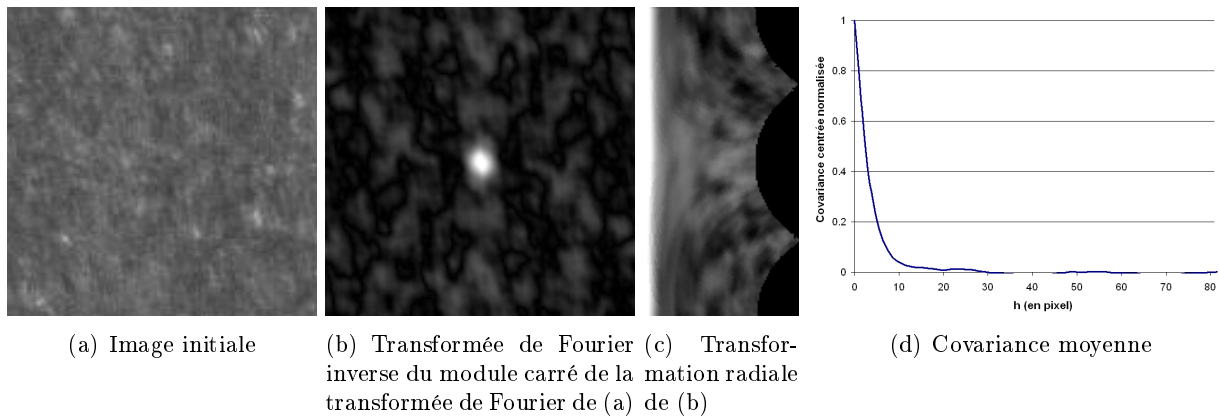


FIGURE 9.5 – Etapes pour générer l'auto-correlation moyenne centrée dans toutes les directions de l'espace.

[158] de la texture. La covariance centrée est calculée en prenant le module carré de l'image dans le domaine fréquentiel puis en repassant dans le domaine spatial [108, 158].

Cette image (figure 9.5(b)) contient donc les corrélations pour tous les angles et pour toutes les distances entre les bi-points. On calcule donc la covariance moyenne en moyennant les valeurs de chaque colonne après avoir réalisé une transformation radiale (voir la figure 9.5(d)). De cette courbe, on extrait **10 descripteurs** qui permettent de caractériser la texture.

#### 9.4.6 Granulométries

La caractérisation d'une texture passe aussi par le calcul d'une granulométrie morphologique. Les granulométries ont été définies à plusieurs reprises dans ce document de thèse (chapitres 2 et 3) et consistent à filtrer l'image avec une série d'ouvertures de tailles différentes. Le pattern spectrum mesure le volume des structures qui ont été supprimées de l'image entre deux ouvertures de tailles successives. On choisira un élément structurant ayant une forme carrée pour réaliser cette étude. On extrait de ce pattern spectrum 4 classes de tailles : de 0 à 5 mm, de 5 mm à 9 mm, de 9 mm à 15 mm et de 15 mm à 19 mm. D'une manière duale, le calcul des anti-granulométries permet de mesurer la taille des structures sombres de l'image. La famille de filtres utilisée est constituée de fermetures morphologiques de différentes tailles. Les mêmes classes sont utilisées que pour les granulométries fournissant au total **8 descripteurs**.

### 9.5 Apprentissage non supervisé

Pour caractériser une texture particulière, un ensemble important de descripteurs doit être utilisé : nous obtenons 50 voir 60 descripteurs de texture. Or si on reprend la stratégie générale que nous souhaitons mettre en place, une analyse statistique non supervisée sera utilisée pour créer une première segmentation des textures présentes dans les images : le masque des textures. Ce masque des textures sera corrigé par l'utilisateur pour obtenir la vérité terrain, utilisée lors de l'apprentissage supervisé. Nous proposons deux méthodes de classification pour séparer automatiquement nos données en  $K$  groupes homogènes.

#### 9.5.1 Les $K$ moyennes mobiles

Les  $K$  moyennes mobiles ou *K-means* en anglais est une méthode de classification qui permet de séparer le nuage de points dans l'espace des descripteurs en  $K$  partitions. Soit  $(x_1, x_2, \dots, x_n)$   $n$  vecteurs de taille  $p$ , des points dans l'espace des descripteurs, et soit une partition  $S = \{S_1, S_2, \dots, S_K\}$  composée de  $K$  classes, on cherche alors à minimiser la somme des carrés des distances avec le centroïde, à l'intérieur de chaque partition :

$$\hat{S} = \operatorname{argmin}_S \sum_{k=1}^K \sum_{x_i \in S_k} \|x_i - \mu_k\|^2 \quad (9.9)$$

avec  $\mu_k$ , la valeur moyenne des points de la partition  $S_k$  que l'on appelle le centroïde numéro  $k$ .

La recherche de la solution optimale de l'équation 9.9 est *NP-difficile* mais des algorithmes heuristiques sont couramment utilisés pour trouver une solution qui converge souvent vers un optimum local.

Des variantes de cet algorithme peuvent être utilisées pour initialiser différemment les moyennes mobiles initiales : Forgy [64], Hartigan et Wong [86], Lloyd [124] ou MacQueen [128]. Toutes ces méthodes ont néanmoins un inconvénient majeur, elles ont tendances à équilibrer les populations des classes, donnant des résultats peu concluants pour nos applications (voir les résultats sur une base de données synthétique “mouse”, figure 9.6(b)). Pour remédier à ce problème, on utilise une seconde méthode, le mélange de gaussiennes entraînée avec un algorithme *Expectation-Maximization* (EM).

#### 9.5.2 Mélange de gaussiennes avec un algorithme EM

Généralement, on maximise (ou minimise) une fonction pour résoudre des problèmes d'optimisation. Il existe de nombreuses techniques, toutes avec des caractéristiques propres, et l'une d'entre elle, le maximum de vraisemblance (likelihood en anglais), consiste à ajuster les variables d'un modèle statistique à partir d'une base de données. Ainsi, le modèle décrit les données et estime les autres.

Cette méthode de classification est basée sur des modèles de probabilité pour estimer les valeurs manquantes avec un algorithme EM, pour maximiser la vraisemblance des variables non observées dans un mélange de gaussiennes.

Un algorithme EM est une méthode non supervisée, itérative et sous optimale, qui essaie de trouver les paramètres de la distribution de probabilité qui maximise la vraisemblance de ces attributs [50]. L'initialisation de cet algorithme est réalisée en utilisant une classification hiérarchique<sup>1</sup>. A chaque

---

1. Classification hiérarchique : à l'initialisation, chaque point représente un groupe et à chaque itération, les deux groupes qui maximise un critère d'homogénéité seront fusionnés [248].

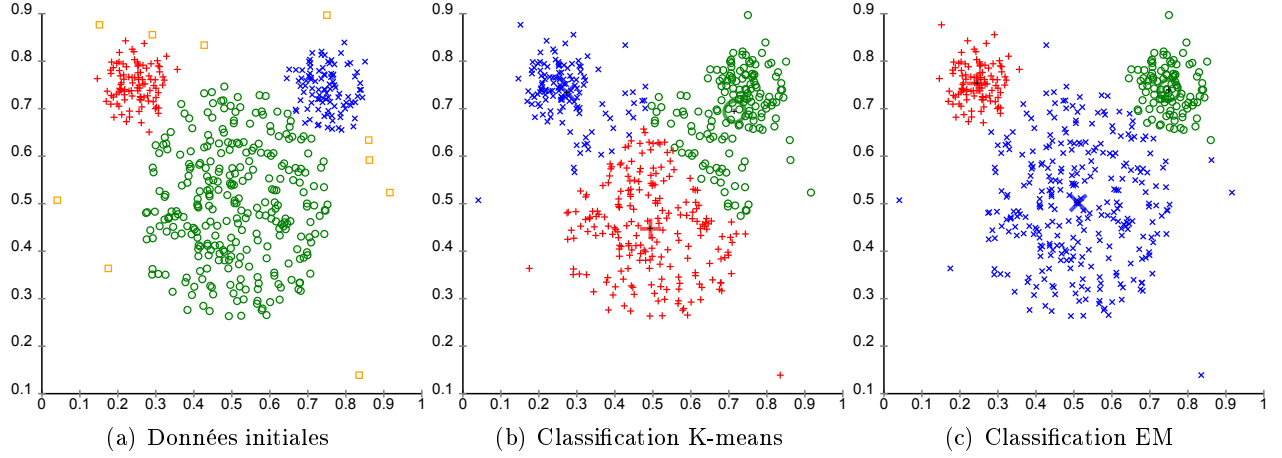


FIGURE 9.6 – Base de données *mouse* synthétique (a) avec les résultats de classification en trois classes avec les méthodes K-means (b) et EM (c).

itération, l'étape *E* consiste à estimer les probabilités d'appartenance des points à chaque groupe, puis l'étape *M* estime le vecteur des paramètres pour chaque groupe. On utilise la méthode de Fraley et Raftery [66] à travers le package R appelé *MClust*<sup>1</sup>.

La figure 9.6(c) présente les améliorations des résultats de classification sur la même base de données *mouse* de la section 9.5.1.

## 9.6 Apprentissage supervisé

L'étape de correction de la segmentation est présentée dans la section 9.7.3 et on considère dans cette section, que la vérité terrain a été validée par un opérateur. On note qu'il peut y avoir au maximum cinq textures différentes à détecter, la texture saine et 4 niveaux de dégradation différents de la texture en arrachement ( $m$  correspond au nombre de textures). Ainsi, on choisit de mettre en place  $m$  apprentissages supervisés pour créer  $m$  modèles statistiques avec une stratégie *un contre tous*. On utilisera donc des régressions logistiques pour obtenir un ensemble de probabilités pour chaque patch. Enfin, l'assignation finale se fera en sélectionnant la classe donnant la plus grande probabilité.

### 9.6.1 Création des bases de données

Dans ce processus, la création des différentes bases de données est une étape importante. Grâce à la vérité terrain, nous connaissons parfaitement le nombre de patches dans chaque classe que l'on note  $n_1, \dots, n_i, \dots, n_m$ . La méthodologie *un contre tous* consiste à créer une base de données avec deux catégories : la classe  $i$  d'une part avec le label 1 et les classes  $j \in \{1, m\}, j \neq i$ , d'autre part avec le label 0. Il faut donc équilibrer nos bases de données et deux situations peuvent se produire :

1. Si  $n_i < \sum_{j \in \{1, m\}, j \neq i} n_j$ , alors on réplique aléatoirement des instances de la classe  $i$  jusqu'à obtenir exactement le même nombre d'instances dans chaque classe.

1. Package R MClust <http://cran.r-project.org/web/packages/mclust>



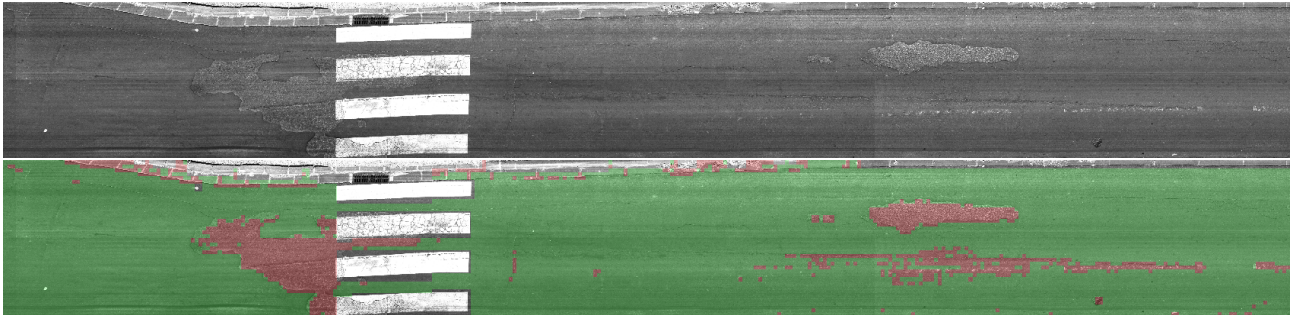


FIGURE 9.7 – Classement d’un secteur de route, superposé aux images originales. L’apprentissage statistique utilise 2 des 6 images (en bas à gauche et en haut à droite) avec les descripteurs basés sur l’histogramme, les LBP et les caractéristiques de Haralick.

TABLEAU 9.1 – Tableau récapitulatif des résultats de classement.

	Texture saine ( $m^2$ )	Texture arrachement ( $m^2$ )
Image globale	105	13
Bandes de roulement	40	6

2. Si  $n_i > \sum_{j \in \{1, m\}, j \neq i} n_j$ , alors on choisit aléatoirement des instances de la classe minoritaire à répliquer.

Nous obtenons  $m$  bases de données équilibrées pour générer nos modèles statistiques.

### 9.6.2 Création des modèles

La régression logistique est une méthode d’apprentissage statistique supervisée que nous avons décrite dans le chapitre 6 et utilisée pour l’application TOCATA du chapitre 7. Cette méthode fournit une probabilité d’appartenir à la classe ayant le label 1 dans la base de données. On calcule donc  $m$  régressions logistiques et pour tous les patches, il suffit de générer les différentes probabilités et d’assigner les patches aux classes ayant données les réponses maximales.

### 9.6.3 Résultats de classement

Chaque nouvelle image est décomposée en une série d’imagettes de taille 100 par 100 pixels et une fois les descripteurs calculés, il est possible d’assigner ces imagettes à des classes précises. La figure 9.7 montre les résultats obtenus et on note deux caractéristiques : les caniveaux ont été classés dans l’une des deux catégories puisqu’ils n’ont pas été détectés automatiquement lors des prétraitements, comme c’est le cas pour les lignes blanches. Tous les éléments externes à la chaussée introduisent des biais dans les résultats. Enfin, on remarque que les patches placés sur une zone qui n’est pas complètement uniforme sont classés en tant qu’arrachement.

Une fois la segmentation obtenue, il est alors aisé de calculer la surface occupée pour chaque type de texture dans les images. Une mesure de la texture en arrachement sera donnée sur l’image entière et sur les bandes de roulement pour générer un indice de dégradation des voies.

La figure 9.8 présente d'autres résultats obtenus avec la même méthodologie. On montre à travers ces différents exemples la généralité de cette approche.

## 9.7 Solution logicielle

Le développement d'une interface homme/machine a permis de faciliter la réalisation de toutes les étapes décrites précédemment dans ce chapitre et rend ce traitement accessible à des personnes non au fait des méthodes d'analyse d'images et d'analyse statistiques. Cette interface graphique propose 4 étapes distinctes.

### 9.7.1 Chargement des images et sélection des régions d'intérêt



Après avoir choisi les images à charger à travers la boîte de dialogue classique de choix multiples de fichiers, des miniatures sont créées avec un facteur 20 de sous-échantillonnage, pour l'affichage et la détection des lignes blanches. Le nom de chaque image est analysé pour reconstituer correctement la route en sa totalité. La détection des lignes blanches est automatique lancée pour masquer ces zones. L'image 9.9 illustre l'interface graphique générale du logiciel après avoir chargé les images.

On note néanmoins la présence d'éléments sur la route non pertinents pour l'analyse des textures : les caniveaux, les bouches d'égout, de l'herbe... Tous ces éléments sont mêmes gênant puisqu'ils possèdent des signatures statistiques très différentes des textures de la route. Il faut donc masquer ces *outliers* manuellement. En cliquant sur une des images initiales, cette dernière est isolée et les régions non pertinentes sont masquées avec un cliquer-déplacer sur l'image.

### 9.7.2 Sélection des descripteurs et analyse statistique non supervisée



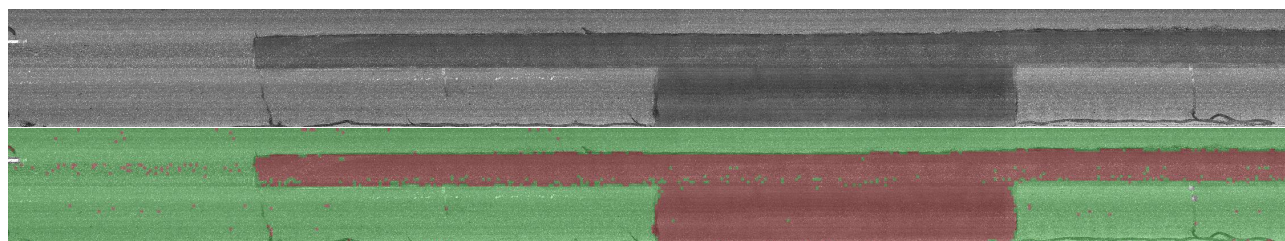
Les sélections des images et des régions d'intérêt terminées, on choisit les descripteurs à calculer sur les images. Une boîte de dialogue permet de cocher et de décocher les méthodes décrites dans la section 9.4 de ce chapitre. Les images sont automatiquement décomposées en patches de taille 100 par 100 pixels, les différents descripteurs sont calculés et une méthode d'analyse statistique non supervisée est lancée pour avoir une première segmentation, qui apparaît dans la partie basse de l'interface graphique (voir la figure 9.10).

Cette première segmentation est en générale jamais parfaite et il faut que l'utilisateur puisse corriger facilement cette segmentation depuis l'interface graphique.

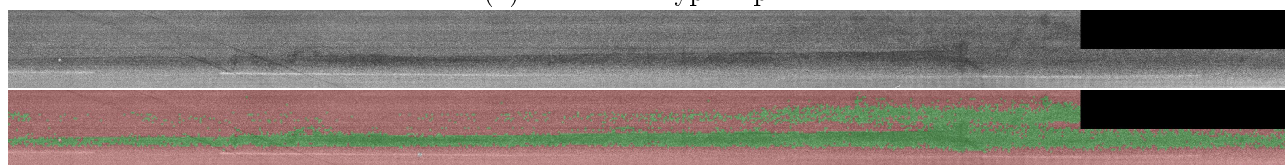
### 9.7.3 Correction de la segmentation

On a imaginé deux moyens différents et complémentaires pour corriger la segmentation. La première méthode consiste à réaliser un cliquer-déplacer sur la segmentation pour changer le label des patches qui ont été classés dans la mauvaise catégorie. La seconde méthode est tout aussi indispensable que la première puisque l'on va pouvoir modifier les labels des points dans l'espace des descripteurs.

Après avoir calculé  $p$  descripteurs, nous obtenons un nuage de points ayant  $p$  dimensions. Ce nuage de points est dans l'état, impossible à afficher sur un écran 2-D. Par conséquent, on réalise une analyse



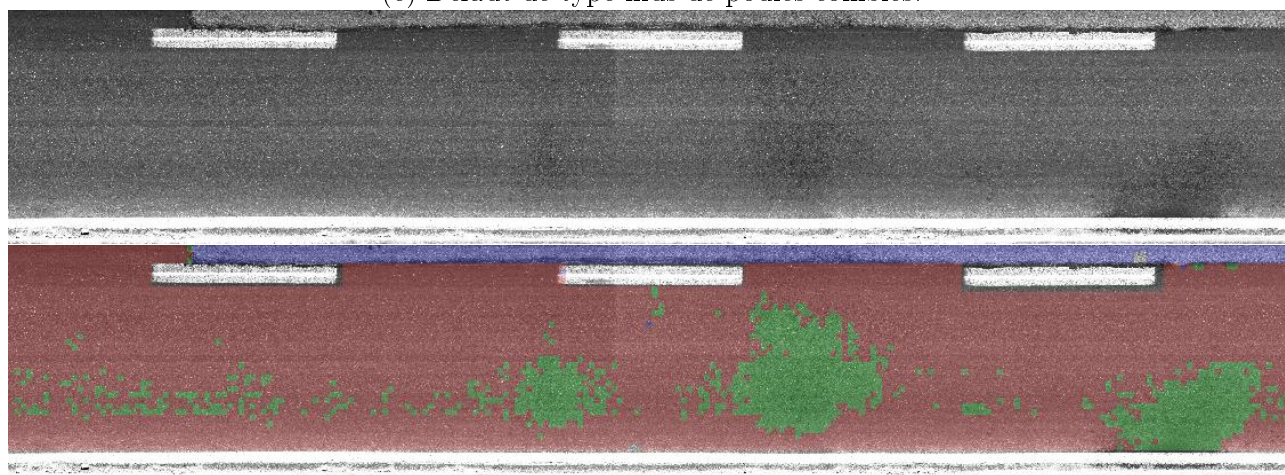
(a) Défaut de type reprise.



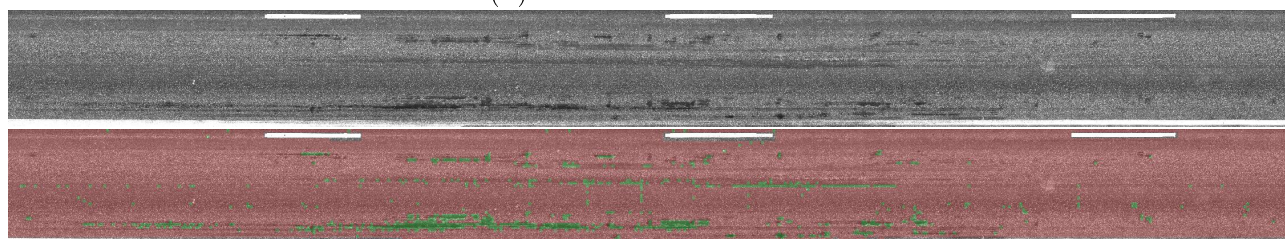
(b) Défaut de type pelade.



(c) Défaut de type nids de poules comblés.



(d) Début d'arrachement.



(e) Défaut de type ressuage.

FIGURE 9.8 – Résultats corrects pour la détection de différents types de défauts de surface.



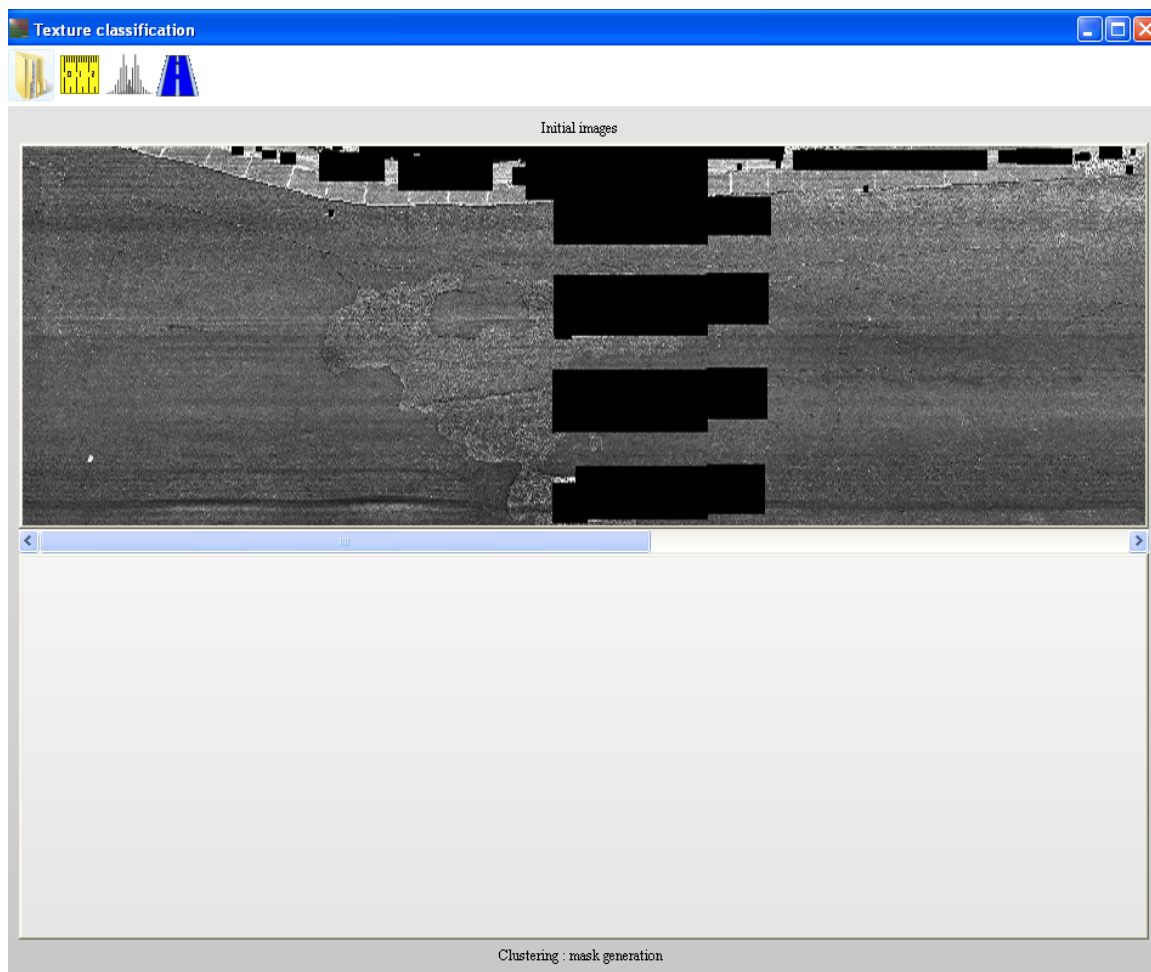


FIGURE 9.9 – L’interface graphique générale est composée de 3 parties distinctes : la barre d’outils pour lancer les traitements, la partie haute de la fenêtre pour afficher l’image initiale et la partie basse pour afficher la vérité terrain.

en composantes principales (ACP) [103, 105, 125] pour déterminer les deux premiers vecteurs propres du nuage afin de réduire l’information et de le projeter dans cette nouvelle base.

La figure 9.10 propose le résultat de l’ACP sur un nuage de points ayant une dimension supérieure à 50. Chaque point correspond à un patch et sa couleur correspond au label attribuée par une méthode de classification non supervisée. Il est alors possible de sélectionner des points dans cette représentation graphique par un cliquer-déplacer pour changer la classe d’affectation de ces patches.

Cette segmentation est corrigée manuellement par deux méthodes très complémentaires en prenant en compte la distance spatiale (corrections sur l’image) et la distance dans l’espace des descripteurs (correction sur l’ACP). On peut donc facilement obtenir la vérité terrain de l’image qui nous servira pour l’étape de classement (apprentissage supervisé).

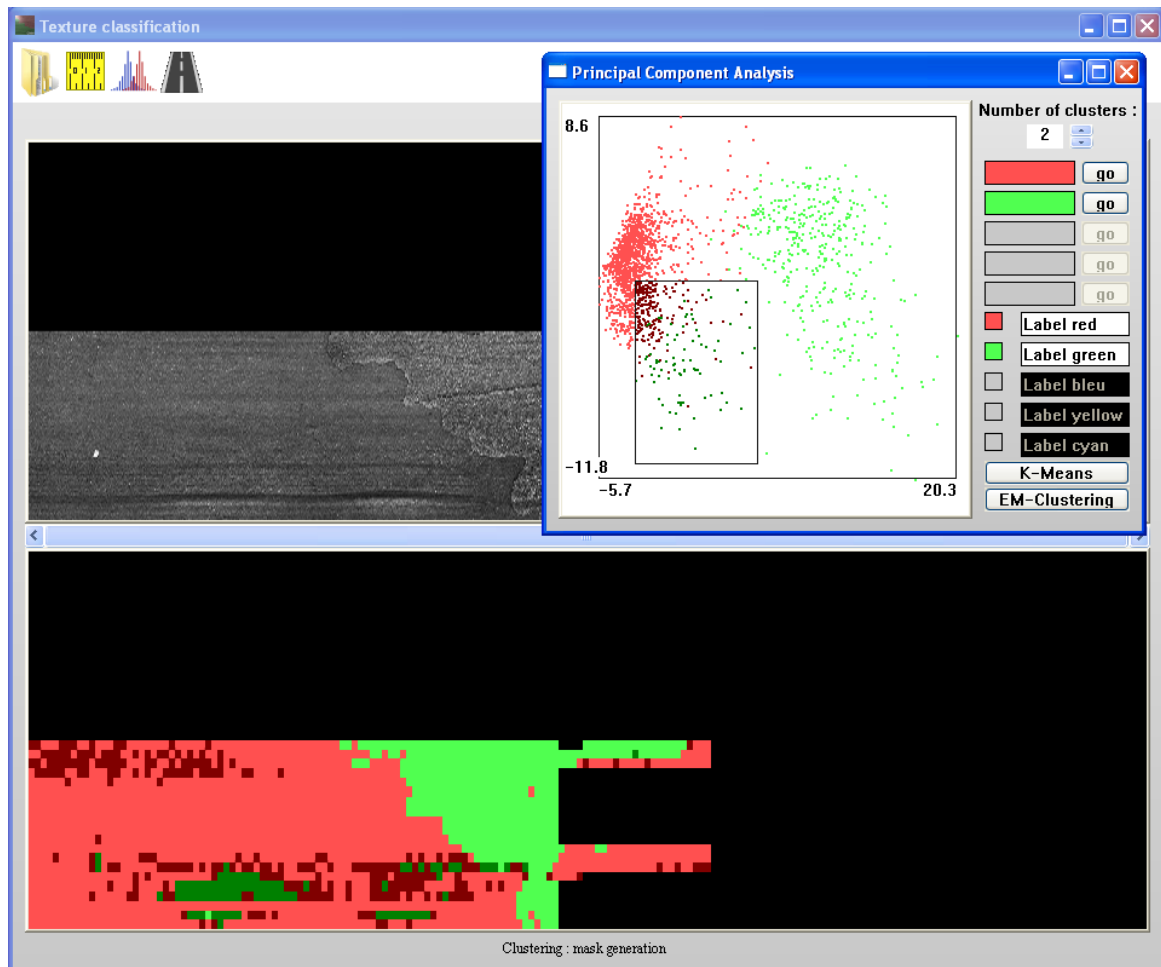


FIGURE 9.10 – Illustration de l’interface graphique principale et de l’analyse en composantes principales pour la correction manuelle de la vérité terrain. La sélection de points sur l’ACP (le rectangle noir) revient à sélectionner des patches sur l’image. Deux méthodes de corrections sont proposées : sur l’ACP et sur la vérité terrain.

### 9.7.4 Analyse statistique supervisée et classement



Un apprentissage statistique est réalisé en suivant la méthodologie *un contre tous* définie dans la section 9.6. Ce logiciel est interfacé avec le logiciel d’analyse statistique R<sup>1</sup> pour générer autant de régressions logistiques que de classes dans la vérité terrain.



Une fois l’apprentissage statistique réalisé, on choisit un répertoire contenant toutes les images à traiter puis nous classons tous les patches issus de ces images. Pour visualiser les résultats, nous superposons les labels aux images initiales donnant la figure 9.7.

Enfin, en même temps que le classement, la surface totale des classes en arrachements est calculée donnant un indicateur sur la dégradation de la route analysée.

1. Le projet R : <http://www.r-project.org/>

## 9.8 Bilan et perspectives

Des capteurs linéaires fixées sur un camion constituent le premier maillon pour réaliser un contrôle non destructif des voies de circulation. Le flot d'images est alors analysé automatiquement afin de détecter les défauts de surface des routes pour quantifier leur dégradation et suivre leur évolution dans le temps. Dans cette étude, on s'intéresse à la détection des arrachements par l'analyse de la texture des images. Il faut donc faire la distinction entre une texture saine et les différentes textures en arrachement. L'existence de nombreux revêtements de surface rend indispensable l'utilisation d'une stratégie générique : descripteurs morphologiques couplés à une étape d'apprentissage statistique.

On propose de nombreux descripteurs afin d'analyser la texture des images : des méthodes basées sur l'histogramme, sur les *local binary patterns*, sur la covariance, sur les matrices de co-occurrence, sur les matrices GLSZM ou sur les granulométries. Cet ensemble de variables décrivent la texture et une analyse statistique sera utilisée pour extraire la signature de chaque classe.

Pour réaliser une analyse statistique supervisée, il faut connaître la vérité terrain de quelques images pour construire une base d'apprentissage. Aussi, on utilise une méthode d'apprentissage non supervisée pour générer une première segmentation : les *K means*. Cette méthode permet de rapidement classer les patches dans les différentes catégories. Cependant, cette méthode a tendance à équilibrer le cardinal de chaque groupe, ce qui peut s'avérer gênant. Ainsi, une seconde méthode de classification est proposée pour résoudre ce problème : les mélanges de gaussiennes avec un algorithme *Expectation-Maximization*. Cette segmentation sera, par la suite, corrigée par l'utilisateur. On construit alors des régressions logistiques, une pour chaque texture à détecter, avec une méthode un contre tous, puis nous classons les images des routes pour extraire les zones en arrachement.

Pour aider l'utilisateur dans la réalisation de toutes ces étapes, un logiciel intuitif est proposé. L'étape de correction de la segmentation, qui aurait pu être fastidieuse sans aide logicielle, est maintenant facilitée. L'utilisateur peut désormais sélectionner les patches à modifier soit dans l'espace spatial, soit dans l'espace des descripteurs, après avoir réalisé une analyse en composantes principales.

Les premiers résultats utilisant cette stratégie sont très prometteurs, puisque différents types de défauts de surface peuvent être détectés sur des routes très différentes les unes des autres. Ainsi, la principale qualité de la stratégie mise en place pour cette étude est sa grande adaptabilité, aussi bien au niveau des défauts que des routes. À notre connaissance, c'est la première fois qu'un système automatisé permet de détecter les défauts de types arrachements, reprises ou pelades.

Une limitation de cette stratégie apparaît lorsque les défauts à détecter sont très petits. En effet, le découpage de l'image peut alors morceler le défaut et le noyer dans la texture de la route. Pour résoudre ce problème, Il faudrait adapter la taille du patch à la taille du défaut à détecter.

Les perspectives de cette étude sont au nombre de trois : des tests plus poussés doivent être réalisés pour finir la validation de cette stratégie, de nouveaux descripteurs peuvent être imaginés pour rendre plus discriminant l'analyse de la texture pour ces images et des régressions logistiques type AdaCOS, chapitre 6, peuvent être utilisées pour sélectionner un ensemble parcimonieux de descripteurs (par groupe) et accélérer les temps de calcul du classement.

## 9. PROJET COLAS

---

L'objectif principal de ce travail consistait à concevoir des méthodes automatiques pour extraire les défauts de surface de pièces métalliques à travers le projet TOCATA. Cependant, les approches classiques en analyse d'images (succession de filtres suivit d'une segmentation) sont inutilisables pour cette application puisqu'il faudrait alors reprendre tout le travail effectué lorsqu'une nouvelle pièce ou un nouveau type de défaut doit être traité. Concevoir une stratégie globale et générique nécessite d'utiliser des méthodes plus complexes associant le traitement d'images et l'analyse statistique. Ainsi, répondre aux contraintes industrielles nous a amené à travailler sur deux axes de recherche principaux, véritables fils conducteurs de cette thèse : la détection et l'extraction des éléments filiformes d'une part et l'optimisation des temps de calcul d'autre part. Aussi, tout au long de ce mémoire, nous avons présenté et développé de nouvelles méthodes dans le cadre de la morphologie mathématique mais aussi dans le cadre de l'apprentissage statistique supervisé. Enfin, des solutions adaptées aux besoins des industriels ont été développées dans le cadre de partenariats avec le CMM (L'Oréal, Colas et le projet TOCATA).

## Contributions de cette thèse

### Contributions en morphologie mathématique

Dans un souci d'optimisation des temps de calcul des opérateurs de base de la morphologie mathématique pour la détection des éléments fins, nous avons introduit un nouvel algorithme permettant de construire des ouvertures linéaires en temps constant par pixel. Cet algorithme est extrêmement flexible puisqu'il permet de calculer des ouvertures 1-D mais aussi des granulométries 1-D, des arbres de composantes connexes 1-D ou des ouvertures par reconstruction 1-D pour des données d'entrée ayant une précision quelconque (entier ou flottant), tout en étant plus rapide que d'autres algorithmes de l'état de l'art (seul celui de Van Droogenbroeck reste le plus rapide pour des ouvertures 1-D sur des images 8-bits).

Cet algorithme 1-D est la base d'un opérateur très prometteur, les ouvertures parcimonieuses par chemins. En comparaison avec les ouvertures par chemins classiques, les PPOs analysent uniquement le sous ensemble de chemins qui porte le plus d'information possible. Différentes stratégies d'extraction de ce sous ensemble sont proposées et fournissent une détection fine des structures filiformes de l'image. On a montré que la précision de ces opérateurs est très bonne et que les temps de calcul sont réduits



## CONCLUSIONS ET PERSPECTIVES

---

de plusieurs ordres de grandeur, surtout lorsque l'on analyse des chemins incomplets. Ces opérateurs sont parcimonieux de deux points de vue : du point de vue des chemins, car seul un sous-ensemble est pris en compte et du point de vue des pixels, car certains ne sont jamais visités.

Pour la détection de structures plus tortueuses dans l'image, on peut utiliser les éléments structurants par croissance de régions REGSEs. Ils font partie de la morphologie mathématique adaptative et ajustent leur forme pour suivre les structures filiformes de l'image à préserver. Des nouveaux opérateurs qui analysent la forme des éléments structurants adaptatifs sont définis pour extraire des éléments structurants, une information différente que le minimum ou le supremum. On peut ainsi extraire les structures filiformes de l'image en calculant l'inertie de chaque REGSE.

Une nouvelle famille d'opérateurs connexes est proposée dans ce document. Les amincissements par attributs géodésiques permettent d'analyser toutes les composantes connexes de l'image et de les filtrer avec différents critères. On a introduit un nouvel attribut géodésique : le diamètre barycentrique, qui donne une mesure de longueur de l'objet. En utilisant cet attribut, les temps de calcul sont considérablement réduits pour devenir plus rapides que ceux des ouvertures par chemins classiques. Pour compléter cette mesure de la longueur d'un objet, on propose aussi des attributs géométriques basés sur l'axe médian de l'image pour mesurer plus précisément la longueur des structures filiformes interconnectées.

### Contribution en analyse statistique

Dans de nombreuses applications, l'analyse statistique devient une discipline indispensable pour compléter l'analyse d'images classique. Dans ce document de thèse, nous avons utilisé de nombreuses méthodes d'analyse statistique : analyse en composantes principales, apprentissage non supervisé, test de normalité, test de significativité et des méthodes d'apprentissage supervisées. C'est dans cette dernière catégorie que l'on retrouve la contribution en analyse statistique. La méthode AdaCOS réalise des régressions linéaires ou logistiques pénalisées et parcimonieuses afin de prendre en compte une fonction de coût que l'on cherche à minimiser. Pour nos applications, on cherche à contraindre les temps de calcul et cette méthode permet de faire un compromis entre la qualité du modèle généré et les temps de calcul.

### Contributions aux projets industriels

Les contributions méthodologiques de ce document de thèse sont intimement liées aux applications industrielles. Le projet TOCATA reprend toutes les nouveautés méthodologiques apportées dans ce document. Les éléments filiformes à détecter sont des défauts de surface de type fissure et on propose une stratégie générique permettant d'extraire différents types de défauts sur différents types de pièces. De plus, un logiciel complet avec une interface graphique spécifique, reprend toutes les étapes du processus pour qu'un utilisateur, sans connaissance approfondie en traitement d'images ou en analyse statistique, puisse utiliser ce logiciel.

Les contributions pour le projet L'Oréal sont au nombre de deux. Premièrement, les descripteurs morphologiques ont été introduits pour caractériser les réseaux fibreux du derme de la peau (analyse de la forme des fibres, de la distribution de taille, de l'orientation et de l'organisation spatiale). Enfin, des méthodes statistiques sont utilisées pour faciliter l'analyse des données obtenues. Un test de significativité sélectif est proposé pour classer et pour visualiser les paramètres qui évoluent le plus avec le temps.

Pour le projet Colas, des opérateurs de traitement d'images ont été utilisés pour réaliser un contrôle non destructif des routes. Ainsi, les défauts de type arrachement sont détectés en analysant la texture

des images. Un logiciel complet a été fourni, permettant de réaliser toutes les étapes de la stratégie mise en place : sélection des images et des régions d'intérêt, choix des descripteurs, apprentissage non supervisé sur des patches de taille 100 par 100 pixels, analyse en composantes principales et correction de la segmentation et enfin, analyse supervisée et quantification des défauts des routes.

## Contributions pour le CMM

Tous les algorithmes développés au cours de cette thèse ont été intégrés à la librairie de traitement d'images du CMM, la librairie *Morph-M*. Voici quelques exemples de méthodes intégrées :

- **ThinStructures** avec les ouvertures par chemins, les ouvertures parcimonieuses par chemins, les amincissements géodésiques et topologiques, les ouvertures 1-D, les  $\lambda$  zones plates, les REGSEs...
- **FreeFFT**, une implémentation de transformée de Fourier rapide libre de droit.
- **NoiseReduction** avec des implémentations rapides du filtre gaussien, du filtre bilatéral, du filtre *meanshift*, des filtres de rang...
- **OptimalEdgeDetection** avec les filtres de détection optimale des contours : algorithmes de Canny et de Deriche.
- **Filtres de Gabor, Haralick, les ouvertures aréolaires rapides, les opérateurs de reconstruction rapides, des squelettes...**

## Perspectives

Cette thèse apporte de nouveaux outils méthodologiques pour la détection des objets fins de l'image et de nombreuses perspectives sont envisageables pour chacune des méthodes développées.

De plus, l'association réussie entre le traitement d'images et les outils statistiques est très prometteur. Ces deux disciplines complémentaires apportent des solutions robustes et génériques d'un point de vue applicatif.

## Perspectives méthodologiques

Les ouvertures parcimonieuses par chemins explorent uniquement un sous ensemble de chemins de l'image, elles sont par conséquent très rapides et présentent de nombreux avantages par rapport aux ouvertures par chemins classiques. Plusieurs extensions de ces opérateurs sont possibles, tant au niveau de la rapidité qu'au niveau de la fonctionnalité. Pour construire des filtres alternés séquentiels parcimonieux sur des chemins (chapitre 3, équations 3.14 et 3.15) d'une manière efficace, il semble possible de construire un algorithme semblable à celui du chapitre 2. Calculer des filtres alternés séquentiels en une unique passe sur le signal permettrait d'accélérer cet opérateur. De la même manière, trouver un algorithme permettant de calculer des ouvertures de rang maximal sur un signal 1-D en temps constant permettra d'obtenir des ouvertures parcimonieuses par chemins robustes aux bruits. Enfin, des pistes très prometteuses consistent à créer d'autres filtres parcimonieux, comme un filtre DoG parcimonieux ou un filtre de Gabor parcimonieux. Il reste aussi à définir une méthode pour calculer le pattern spectrum de l'image sans avoir à calculer toutes les ouvertures parcimonieuses par chemins de tailles successives. Enfin, l'extraction des chemins peut être étendue en utilisant une méthode ayant un comportement intermédiaire entre les chemins localement maximaux et les chemins globalement optimaux.

Les REGSEs sont des éléments structurants adaptatifs et leur forme fournit une information importante sur les structures sous jacentes. Il est néanmoins possible d'adapter les opérateurs qui analysent

## CONCLUSIONS ET PERSPECTIVES

---

la forme des REGSEs à d'autres éléments structurants adaptatifs. Avec cette méthodologie, on analyse la forme des  $\lambda$ -zones plates pour extraire d'une manière très efficace, les structures filiformes de l'image.

L'extraction des structures fines ayant une tortuosité quelconque est rendue possible grâce à l'utilisation des filtres connexes. Cependant, les structures longues et filiformes sont très sensibles au bruit et si elles sont déconnectées, alors leur longueur est sous estimée. Étendre la connectivité de ces filtres avec la connectivité de seconde génération [174] permet d'améliorer leur robustesse.

### Perspectives applicatives

D'un point de vue applicatif, des études complémentaires peuvent être effectuées pour ajouter de nouvelles contributions aux différents projets réalisés au cours de cette thèse.

Pour le projet TOCATA, les descripteurs ont été principalement adaptés aux défauts de type fissure. Il faut donc ajouter des opérateurs plus adaptés à la recherche d'autres défauts de surface, comme les piqûres ou les brûlures. Il est aussi nécessaire de travailler sur les prétraitements des images afin de les uniformiser. C'est un point très important qui n'a pas pu être abordé au cours de cette étude. En effet, le démonstrateur et le système optique ont été mis au point dans les derniers mois du projet et, par conséquent, l'étape de validation doit être largement approfondie pour définir les performances globales du système. Enfin, on pourra travailler sur l'analyse statistique pour proposer des frontières de séparation qui ne sont plus linéaires.

Pour la quantification des fibres de collagène et des fibres d'élastine du derme de la peau, des outils d'analyse d'images ont été mis en place. Ils pourront donc être utilisés dans de nouvelles études pour visualiser l'impact du soleil sur la peau.

Enfin, la collaboration avec Colas pour le contrôle non destructif des routes a abouti à l'élaboration d'un logiciel complet d'analyse de textures. Ce logiciel peut être amélioré en ajoutant de nouveaux descripteurs qui analysent la texture des images et en utilisant des modèles statistiques plus élaborés (parcimonieux par groupe) pour accélérer les temps de calcul.

## Troisième partie

# Annexes confidentielles



- [1] H. Altendorf. *3D Morphological analysis and modeling of random fiber networks : applied on glass fiber reinforced composites*. Thèse de Doctorat, École des Mines de Paris (MINES-ParisTech), 2011. [14](#)
- [2] H. Altendorf, E. Decenci re, D. Jeulin, P. Peixoto, A. Deniset-Bresseau, E. Angelini, G. Mosser, and M.C. Schanne-Klein. Imaging and 3D morphological analysis of collagen fibrils. *Journal of Microscopy*, 2012. [145](#)
- [3] H. Altendorf and D. Jeulin. 3d directional mathematical morphology for analysis of fiber orientations. *Image Analysis & Stereology*, 28(3) :143–153, 2011. [14](#)
- [4] T.W. Anderson. On the distribution of the two-sample cramer-von mises criterion. *The Annals of Mathematical Statistics*, pages 1148–1159, 1962. [155](#)
- [5] B. Appleton and H. Talbot. Efficient path openings and closings. *Mathematical Morphology : 40 Years On*, pages 33–42, 2005. [44](#), [46](#)
- [6] A. Armagan and R. Zaretzki. Adaptive ridge selector (ARiS). Technical report, Cornell University arXiv, 2008. [129](#)
- [7] D. Attali, J.D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes a state-of-the-art report. *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 109–125, 2009. [110](#)
- [8] F.R. Bach. Bolasso : model consistent lasso estimation through the bootstrap. In *Machine learning, Proceedings of the 25th International Conference on*, pages 33–40. ACM, 2008. [129](#)
- [9] T. Baldeweck, A.M. Pena, E. Tancre de-Bohin, E. Decenci re, P. Dokl dal, S. Koudoro, V. Morard, and F. Meyer. Proc d  pour caract riser l piderme et le derme   partir d images muliphoton tridimensionnelles in vivo de la peau. Brevet, 1161616. 12 2011. [144](#)
- [10] T. Baldeweck, A.M. Pena, E. Tancre de-Bohin, E. Decenci re, and S. Koudoro. Proc d  non invasif de d tection, de visualisation et/ou de quantification 3D sp cifiques d un fluorophore endog ne tel que la m lanine au sein d un tissu biologique. Brevet, 1160172. 11 2011. [145](#)
- [11] T. Baldeweck, E. Tancre de-Bohin, P. Dokl dal, S. Koudoro, V. Morard, F. Meyer, E. Decenci re, and A.M. Pena. Imagerie muliphoton de la peau humaine in vivo. In *Diagnostic et imagerie optique en m decine*, 2012. [144](#)
- [12] T. Baldeweck, E. Tancre de-Bohin, P. Dokl dal, S. Koudoro, V. Morard, F. Meyer, E. Decenci re, and A.M. Pena. In vivo muliphoton microscopy associated to 3D image processing for human skin characterization. In *Multiphoton Microscopy in the Biomedical Sciences*, volume 8226. Photonics West, 2012. [144](#), [145](#)
- [13] J. Bartovsk y, P. Dokl dal, E. Dokl dalov , and M. Bilodeau. Fast streaming algorithm for 1-d morphological opening and closing on 2-d support. In *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 6671 of *LNCS*, pages 296–305. Springer, 2011. [19](#), [36](#)

## BIBLIOGRAPHIE

---

- [14] R. Bellman. Dynamic programming. *Science*, 153(3731) :34, 1966. [49](#)
- [15] J. Berkson. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, pages 357–365, 1944. [135](#)
- [16] L. Bertelli and B.S. Manjunath. Edge preserving filters using geodesic distances on weighted orthogonal domains. In *Image Processing, IEEE International Conference on*, volume 1, pages 317–321. IEEE, 2007. [74](#)
- [17] G. Bertrand. On topological watersheds. *Journal of Mathematical Imaging and Vision*, 22(2) :217–230, 2005. [14](#), [20](#)
- [18] S. Beucher. *Segmentation d'image et morphologie mathématique*. Thèse de Doctorat, École des Mines de Paris, 1990. [14](#), [48](#)
- [19] S. Beucher. Numerical residues. *Image and Vision Computing*, 25(4) :405–415, 2007. [123](#)
- [20] S. Beucher, J.M. Blosseville, and F. Lenoir. Traffic spatial measurements using video image processing. *Intelligent Robots and Computer Vision, Proceedings SPIE*, 848 :648–655, 1987. [74](#)
- [21] S. Beucher and F. Meyer. The morphological approach to segmentation : the watershed transformation. *Optical engineering*, 34 :433–481, 1992. [146](#)
- [22] H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, 19(5) :362–380, 1967. [110](#)
- [23] N. Bouaynaya, M. Charif-Chefchaoui, and D. Schonfeld. Theoretical foundations of spatially-variant mathematical morphology part I : Binary images. *Pattern Analysis and Machine Intelligence, IEEE transactions on*, 30(5) :823, 2008. [74](#)
- [24] N. Bouaynaya and D. Schonfeld. Theoretical foundations of Spatially-Variant mathematical morphology part II : Gray-Level images. *Pattern Analysis and Machine Intelligence, IEEE transactions on*, 30(5) :837–850, 2008. [74](#)
- [25] R.N. Bracewell. *The Fourier transform & its applications*, volume 3. McGraw-Hill, 2000. [9](#), [147](#)
- [26] U.M. Braga-Neto. Alternating sequential filters by Adaptive-Neighborhood structuring functions. In *Mathematical Morphology and Its Applications to Image and Signal Processing, Proceedings of the 3rd International Symposium on*, pages 139–146. Springer-Verlag, May 1996. [74](#)
- [27] U.M. Braga-Neto and J. Goutsias. A theoretical tour of connectivity in image processing and analysis. *Journal of Mathematical Imaging and Vision*, 19(1) :5–31, 2003. [123](#)
- [28] M.C. Branchet, S. Boisnic, C. Frances, and A.M. Robert. Skin thickness changes in normal aging skin. *Gerontology*, 36(1) :28–35, 1990. [145](#)
- [29] E.J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64 :377–389, 1996. [19](#), [98](#), [100](#), [119](#)
- [30] J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1) :25–30, 1965. [9](#), [29](#)
- [31] M. Buckley and J. Yang. Regularised shortest-path extraction. *Pattern Recognition Letters*, 18(7) :621–629, 1997. [49](#), [57](#)
- [32] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8(6) :679–698, 1986. [14](#)
- [33] S. Chambon and J.M. Moliard. Automatic road pavement assessment with image processing : Review and comparison. *International Journal of Geophysics*, 989354 :20, 2011. [166](#)
- [34] J. Chaussard, M. Couprie, and H. Talbot. A discrete  $\lambda$ -medial axis. In *Discrete Geometry for Computer Imagery, Proceedings of the 15th IAPR International Conference on*, pages 421–433. Springer-Verlag, 2009. [110](#)
- [35] F. Chazal and A. Lieutier. The  $\lambda$ -medial axis. *Graphical Models*, 67(4) :304–331, 2005. [110](#)
- [36] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1) :129–159, 2001. [128](#)

- [37] F. Cheng and A.N. Venetsanopoulos. An adaptive morphological filter for image processing. *Image Processing, IEEE Transactions on*, 1(4) :533–539, 1992. 74
- [38] G. Chinga-Carrasco, A. Miettinen, C.L. Luengo Hendriks, E.K. Gamstedt, and M. Kataja. *Structural Characterisation of Kraft Pulp Fibres and Their Nanofibrillated Materials for Biodegradable Composite Applications in Nanocomposites and Polymers with Analytical Methods*. Intech, 2011. 44
- [39] C. Clienti, M. Bilodeau, and S. Beucher. An efficient hardware architecture without line memories for morphological image processing. In *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 147–156. Springer-Verlag, 2008. 19, 36
- [40] D. Coltuc and I. Pitas. On fast running max-min filtering. *Circuits and Systems II : Analog and Digital Signal Processing, IEEE Transactions on*, 44(8) :660–663, 1997. 19
- [41] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5) :603–619, 2002. 74
- [42] nVidia Corporation. CUDA C Programming Guide, Sep 2010. 41
- [43] nVidia Corporation. NVIDIA GPU Computing Developer Home Page, Jun 2011. 41
- [44] M. Couprie, L. Najman, and G. Bertrand. Quasi-linear algorithms for the topological watershed. *Journal of Mathematical Imaging and Vision*, 22(2) :231–249, 2005. 20
- [45] O. Cuisenaire. Locally adaptable mathematical morphology. In *Image Processing, IEEE International Conference on*, 2005. 74
- [46] R. Dardenne and M. Van Droogenbroeck. libmorpho, [http : //www.ulg.ac.be/telecom/research.html](http://www.ulg.ac.be/telecom/research.html). 36
- [47] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *Information Theory, IEEE Transactions on*, 36(5) :961–1005, 1990. 9
- [48] J. Debayle and J.C. Pinoli. General adaptive neighborhood image processing. *Journal of Mathematical Imaging and Vision*, 25(2) :245–266, 2006. 74
- [49] E. Decenci re, P. Dokl dal, S. Koudoro, E. Tancre de-Bohin, A.M. Pena, and T Baldeweck. Segmentation of three-dimensional in vivo multiphoton images of human skin. *Skin Research and Technology*, 2013. 146, 158
- [50] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977. 174
- [51] R. Deriche. Using Canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2) :167–187, 1987. 14
- [52] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959. 49
- [53] E. Dimitrow, M. Ziemer, M.J. Koehler, J. Norgauer, K. K nig, P. Elsner, and M. Kaatz. Sensitivity and specificity of multiphoton laser tomography for in vivo and ex vivo diagnosis of malignant melanoma. *Journal of Investigative Dermatology*, 129(7) :1752–1758, 2009. 145
- [54] P. Dokl dal. *Grey-Scale Image Segmentation : A Topological Approach*. Th se de Doctorat, Marne-la-Valle University, January 2000. 154
- [55] P. Dokl dal, C. Lohou, L. Perroton, and G. Bertrand. A new thinning algorithm and its application to extraction of blood vessels. In *Biology, Medicine and Biomedical Engineering Modelling and Simulation*, pages 32–37, 1999. 14
- [56] R.O. Duda and P.E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the Association for the Computing Machinery*, 15(1) :11–15, 1972. 14
- [57] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2) :407–451, 2004. 129, 130, 135
- [58] H.A. El Madani, E. Tancre de-Bohin, A. Bensussan, A. Colonna, A. Dupuy, M. Bagot, and A.M. Pena. In vivo multiphoton imaging of human skin : assessment of topical corticosteroid-induced epidermis atrophy and depigmentation. *Journal of Biomedical Optics*, 17, 2012. 145



## BIBLIOGRAPHIE

---

- [59] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456) :1348–1360, 2001. [129](#)
- [60] R.A. Ferguson, D.N. Pratt, P.R. Turtle, I.B. Macintyre, D.P. Moore, P.D. Kearney, M.J. Best, J.L. Gardner, M. Berman, and M.J. Buckley. Road pavement deterioration inspection system. Brevet, 6615648. 9 2003. [166](#)
- [61] G.J. Fisher, S. Kang, J. Varani, Z. Bata-Csorgo, Y. Wan, S. Datta, and J.J. Voorhees. Mechanisms of photoaging and chronological skin aging. *Archives of dermatology*, 138(11) :1462, 2002. [66](#), [160](#)
- [62] R.A. Fisher. *Statistical Methods for Research Workers*, volume 1. Edinburgh : Oliver and Boyd, 1925. [155](#)
- [63] R.A. Fisher. Tests of significance in harmonic analysis. *Proceedings of the Royal Society of London. Series A*, 125(796) :54–59, 1929. [156](#)
- [64] E.W. Forgy. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, 21 :768–769, 1965. [174](#)
- [65] J.B.J. Fourier. *Théorie analytique de la chaleur*, volume 1. Gauthier-Villars et fils, 1888. [147](#)
- [66] C. Fraley and A.E. Raftery. Mclust version 3 for R : Normal mixture modeling and model-based clustering. Technical report, University of Washington, 2006. [175](#)
- [67] A. Frangi, W. Niessen, K. Vincken, and M. Viergever. Multiscale vessel enhancement filtering. *Medical Image Computing and Computer Assisted Intervention*, pages 130–137, 1998. [12](#)
- [68] I.E. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, pages 109–135, 1993. [129](#)
- [69] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1) :1–22, 2010. [135](#)
- [70] D. Gabor. Theory of communication. part 1 : The analysis of information. *Electrical Engineers-Part III : Radio and Communication Engineering, Journal of the Institution of*, 93(26) :429–441, 1946. [11](#)
- [71] G. Gallo and S. Pallottino. Shortest path algorithms. *Annals of Operations Research*, 13(1) :1–79, 1988. [49](#)
- [72] M.M. Galloway. Texture analysis using grey level run lengths. *NASA STI/Recon Technical Report N*, 75 :18555, 1974. [172](#)
- [73] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(1) :1–14, 1996. [12](#)
- [74] D.Z. Gevorkian, J.T. Astola, and S.M. Atourian. Improving Gil-Werman algorithm for running min and max filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5) :526–529, 1997. [19](#)
- [75] S. Ghosh. Adaptive elastic net : An improvement of elastic net to achieve oracle properties. *Preprint*, pages 1–26, 2007. [129](#)
- [76] J.Y. Gil and R. Kimmel. Efficient dilation, erosion, opening, and closing algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12) :1606–1617, 2002. [19](#)
- [77] J.Y. Gil and M. Werman. Computing 2-d min, median, and max filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(5) :504–507, 1993. [19](#), [36](#)
- [78] R. Gordon and R.M. Rangayyan. Feature enhancement of film mammograms using fixed and adaptive neighborhoods. *Applied Optics*, 23(4) :560–564, 1984. [74](#)
- [79] L. Grady and M.P. Jolly. Weights and topology : A study of the effects of graph construction on 3D image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, pages 153–161, 2008. [146](#)
- [80] J. Grazzini and P. Soille. Improved morphological interpolation of elevation contour data with generalised geodesic propagations. volume 4673, pages 742–750. Springer-Verlag, 2007. [102](#)
- [81] J. Grazzini and P. Soille. Adaptive morphological filtering using similarities based on geodesic time. In *Discrete Geometry for Computer Imagery*, pages 519–528. Springer, 2008. [74](#)

- [82] Khronos Group. OpenCL, 2011. [41](#)
- [83] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3 :1157–1182, 2003. [128](#), [129](#)
- [84] A. Haar. *Zur Theorie der Orthogonalen Funktionensysteme*. Thèse de Doctorat, University of Göttingen, 1910. [9](#)
- [85] R.M. Haralick, K. Shanmugam, and I.H. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6) :610–621, 1973. [171](#), [172](#)
- [86] J.A. Hartigan and M.A. Wong. Algorithm AS 136 : A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1) :100–108, 1979. [174](#)
- [87] T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning : data mining, inference, and prediction. volume 2*. Springer Verlag, 2009. [126](#), [128](#), [129](#), [133](#)
- [88] H.J.A.M. Heijmans. Morphological image operators. *Advances in Electronics and Electron Physics*, 1, 1994. [9](#), [19](#)
- [89] H.J.A.M. Heijmans, M. Buckley, and H. Talbot. Path openings and closings. *Probability, Networks and Algorithms*, 1(E 0403) :1–21, 2004. [44](#)
- [90] H.J.A.M. Heijmans, M. Buckley, and H. Talbot. Path openings and closings. *Journal of Mathematical Imaging and Vision*, 22(2) :107–119, 2005. [4](#), [12](#), [44](#), [52](#), [115](#)
- [91] J. Hernández and B. Marcotegui. Ultimate attribute opening segmentation with shape information. *Mathematical Morphology and Its Application to Signal and Image Processing*, pages 205–214, 2009. [123](#)
- [92] W.H. Hesselink. Salembier’s min-tree algorithm turned into breadth first search. *Information Processing Letter*, 88(5) :225–229, 2003. [118](#)
- [93] W.H. Hesselink and J.B.T.M. Roerdink. Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(12) :2204–2217, 2008. [110](#)
- [94] T. Hesterberg, N.H. Choi, L. Meier, and C. Fraley. Least angle and l1 penalized regression : A review. *Statistics Surveys*, 2(2008) :61–93, 2008. [128](#)
- [95] A.E. Hoerl and R.W. Kennard. Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, pages 55–67, 1970. [127](#)
- [96] P.V.C. Hough. Method and means for recognizing complex patterns. Brevet, 3069654. 12 1962. [14](#)
- [97] C.M. Jarque and A.K. Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3) :255–259, 1980. [155](#)
- [98] D. Jeulin. *Modèles Morphologiques de Structures Aléatoires et de Changement d’Échelle*. Thèse de Doctorat d’Etat, Université de Caen, 1991. [61](#)
- [99] D. Jeulin. Modelling random media. *Image Analysis and Stereology*, 21 :S31–S40, 2002. [61](#)
- [100] D. Jeulin and M. Moreaud. Segmentation of 2d and 3d textures from estimates of the local orientation. *Image Analysis and Stereology*, 27 :183–192, 2008. [14](#)
- [101] R. Jones. Component trees for image filtering and segmentation. In *Nonlinear Signal and Image Processing, IEEE Workshop on*. Mackinac Island, 1997. [20](#)
- [102] B.L. Justusson. Median filtering : Statistical properties. *Two-Dimensional Digital Signal Processing II*, pages 161–196, 1981. [9](#)
- [103] Pearson K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559–572, 1901. [179](#)
- [104] P. Karas, V. Morard, J. Bartovský, T. Grandpierre, E. Dokládalová, P. Matula, and P. Dokládal. GPU Implementation of Linear Morphological Openings with Arbitrary Angle. *Journal of Real-Time Image Processing*, 2012. [18](#), [41](#)

## BIBLIOGRAPHIE

---

- [105] K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37. Universitat Helsinki, 1947. [179](#)
- [106] J.C. Klein. *Conception et réalisation d'une unité logique pour l'analyse quantitative d'images*. Thèse de Doctorat, École des Mines de Paris, 1976. [98](#)
- [107] S. Klemenjak, B. Waske, S. Valero, and J. Chanussot. Automatic detection of rivers in high-resolution SAR data. *Selected Topics in Applied Earth Observation and Remote Sensing, IEEE Journal of*, 5(5), 2012. [44](#)
- [108] K. Koch, J. Ohser, and K. Schladitz. Spectral theory for random closed sets and estimating the covariance via frequency space. *Advances in Applied Probability*, pages 603–613, 2003. [173](#)
- [109] M.J. Koehler, K. König, P. Elsner, R. Bückle, and M. Kaatz. In vivo assessment of human skin aging by multiphoton laser scanning tomography. *Optics letters*, 31(19) :2879–2881, 2006. [145](#)
- [110] M.J. Koehler, A. Preller, N. Kindler, P. Elsner, K. König, R. Bückle, and M. Kaatz. Intrinsic, solar and sunbed-induced skin aging measured in vivo by multiphoton laser tomography and biophysical methods. *Skin Research and Technology*, 15(3) :357–363, 2009. [145](#)
- [111] K. König, A. Ehlers, F. Stracke, and I. Riemann. In vivo drug screening in human skin using femtosecond laser multiphoton tomography. *Skin pharmacology and physiology*, 19(2) :78–88, 2006. [145](#)
- [112] R.S. Kurban and J. Bhawan. Histologic changes in skin associated with aging. *The Journal of dermatologic surgery and oncology*, 16(10) :908, 1990. [145](#)
- [113] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. Digital processing of biomedical images, 1976. [74](#)
- [114] C. Lantuéjoul. On the estimation of mean values in individual analysis of particles. *Microscopica Acta*, 5 :266–273, 1980. [33](#)
- [115] C. Lantuejoul and S. Beucher. On the use of the geodesic metric in image analysis. *Journal of Microscopy*, 121(1) :39–49, 1981. [101](#)
- [116] C. Lantuéjoul and F. Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern Recognition*, 17(2) :177–187, 1984. [101](#), [102](#), [103](#), [104](#), [105](#), [151](#)
- [117] R. Lerallut. *Modélisation et interprétation d'images à l'aide de graphes*. Thèse de Doctorat, École des Mines de Paris, MINES ParisTech, 2006. [77](#)
- [118] R. Lerallut, E. Decencièrè, and F. Meyer. Image filtering using morphological amoebas. *Mathematical Morphology : 40 Years On*, pages 13–22, 2005. [74](#)
- [119] R. Lerallut, E. Decencièrè, and F. Meyer. Image filtering using morphological amoebas. *Image and Vision Computing*, 25(4) :395–404, 2007. [74](#)
- [120] D. Lesage, J. Darbon, and C. Burak Argül. An efficient algorithm for connected attribute thinnings and thickenings. In *International Symposium on Visual Computing*, volume 4292, pages 393–404. Springer, 2006. [118](#)
- [121] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. In *Communications of the Association for the Computing Machinery Transactions on Graphics*, volume 23, pages 303–308. ACM, 2004. [146](#)
- [122] H.W. Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, pages 399–402, 1967. [155](#)
- [123] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. In *Computer Vision and Pattern Recognition, International Conference on*, pages 465–470. IEEE, 1996. [14](#)
- [124] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2) :129–137, 1982. [174](#)
- [125] M. Loeve. Probability theory, vol. ii. *Graduate texts in mathematics*, 46 :0–387, 1978. [179](#)
- [126] C.L. Luengo Hendriks. Constrained and dimensionality-independent path openings. *Image Processing, IEEE Transactions on*, 19(6) :1587–1595, 2010. [44](#), [46](#), [57](#), [122](#)

- [127] C.L. Luengo Hendriks and L.J. Van Vliet. Using line segments as structuring elements for sampling-invariant measurements. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11) :1826–1831, 2005. [31](#), [57](#)
- [128] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Mathematical Statistics and Probability, Proceedings of the 5th Berkeley Symposium on*, volume 1, pages 281–297. California, USA, 1967. [174](#)
- [129] G. Malandain and S. Fernández-Vidal. Euclidean skeletons. *Image and Vision Computing*, 16(5) :317–327, 1998. [110](#)
- [130] S.G. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7) :674–693, 1989. [9](#)
- [131] P. Maragos. Pattern spectrum and multiscale shape representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7) :701–716, 1989. [20](#), [21](#), [29](#), [31](#), [153](#)
- [132] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167) :187–217, 1980. [11](#), [14](#)
- [133] B.R. Masters, P.T. So, and E. Gratton. Multiphoton excitation fluorescence microscopy and spectroscopy of in vivo human skin. *Biophysical journal*, 72(6) :2405–2412, 1997. [145](#)
- [134] P. Matas, E. Dokládlová, M. Akil, V. Georgiev, and M. Poupa. Parallel hardware implementation of connected component tree computation. In *Field Programmable Logic and Applications (FPL), International Conference on*, pages 64–69, 2010. [42](#)
- [135] P. Matas, E. Dokládlová, M. Akil, T. Grandpierre, L. Najman, M. Poupa, and V. Georgiev. Parallel algorithm for concurrent computation of connected component tree. In *Advanced Concepts for Intelligent Vision Systems*, pages 230–241. Springer, 2008. [42](#)
- [136] G. Matheron. *Eléments pour une théorie des milieux poreux*. Masson Paris, 1967. [19](#)
- [137] G. Matheron. *Random sets and integral geometry*. Wiley series in probability and mathematical statistics. Wiley, New York, 1974. [19](#), [21](#), [80](#)
- [138] D. McFadden. Conditional logit analysis of qualitative choice behavior. Technical report, University of California, Berkeley. Institute of Urban & Regional Development, 1973. [135](#)
- [139] L. Meier, S. Van De Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 70(1) :53–71, 2008. [137](#)
- [140] N. Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1) :374–393, 2007. [129](#)
- [141] J.D. Mendiola-Santibañez, I.R. Terol-Villalobos, G. Herrera-Ruiz, and A. Fernández-Bouzas. Morphological contrast measure and contrast enhancement : One application to the segmentation of brain MRI. *Signal Processing*, 87(9) :2125–2150, 2007. [95](#)
- [142] D. Menotti, L. Najman, and A. de Albuquerque Araújo. 1D component tree in linear time and space and its application to gray-level image multithresholding. In *Mathematical Morphology, 8th International Symposium on*, pages 437–448. Springer-Verlag, 2007. [20](#)
- [143] F. Meyer. Levelings, image simplification filters for segmentation. *Journal of Mathematical Imaging and Vision*, 20(1) :59–72, 2004. [95](#)
- [144] F. Meyer and P. Maragos. Morphological scale-space representation with levelings. *Scale-Space Theories in Computer Vision*, pages 187–198, 1999. [95](#)
- [145] F. Meyer and P. Maragos. Nonlinear scale-space representation with morphological levelings. *Journal of Visual Communication and Image Representation*, 11(2) :245–265, 2000. [95](#)
- [146] R.E. Miles. On the elimination of edge effects in planar sampling. *Stochastic geometry : a tribute to the memory of Rollo Davidson*, pages 228–247, 1974. [33](#)
- [147] T. Miyatake, M. Ejiri, and H. Matsushima. A fast algorithm for maximum-minimum image filtering. *Systems and Computers in Japan*, 27(13) :74–85, 1996. [19](#)

## BIBLIOGRAPHIE

---

- [148] E.W. Montroll and G.H. Weiss. Random walks on lattices. ii. *Journal of Mathematical Physics*, 6 :167, 1965. [63](#)
- [149] A. Morales. Adaptive Structuring Element for Noise and Artifact Removal. In *Information Sciences and Systems, Proceedings of the Conference on*, 1989. [74](#)
- [150] V. Morard, E. Decenci re, and P. Dokl dal. Geodesic attributes thinnings and thickenings. In *Mathematical Morphology and its Applications to Image and Signal Processing, Proceedings of the 10th International Conference on*, volume 6671 of *LNCS*, pages 200–211. Springer-Verlag, 2011. [98](#)
- [151] V. Morard, E. Decenci re, and P. Dokl dal. Region growing structuring elements and new operators based on their shape. In *Signal and Image Processing, International Conference on*. ACTA Press, 2011. [74](#)
- [152] V. Morard, E. Decenci re, and P. Dokl dal. Efficient geodesic attribute thinnings based on the barycentric diameter. *Journal of Mathematical Imaging and Vision*, 2012. [98](#)
- [153] V. Morard, E. Decenci re, and P. Dokl dal. Characterization of 3D fibrous media with geodesic methods. In *3-D Microstructure Meeting*, 2011. [144](#)
- [154] V. Morard, E. Decenci re, and P. Dokl dal. M thodes g odesiques pour la caract risation des milieux fibreux. In *34 me journ e ISS France*, 2011. [144](#)
- [155] V. Morard, P. Dokl dal, and E. Decenci re. One-dimensional openings, granulometries and component trees in  $O(1)$  per pixel. *Selected Topics in Signal Processing, IEEE Journal of*, 2012. [18](#)
- [156] V. Morard, P. Dokl dal, and E. Decenci re. Linear openings in arbitrary orientation in  $O(1)$  per pixel. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, pages 1457–1460. IEEE, 2011. [18](#)
- [157] M. Moreaud, D. Jeulin, V. Morard, and R. Revel. TEM image analysis and modelling : application to boehmite nanoparticles. *Journal of Microscopy*, 2011. [74](#)
- [158] M. Moreaud, R. Revel, D. Jeulin, and V. Morard. Size of boehmite nanoparticles by TEM images analysis. *Image Analysis and Stereology*, 28 :187–193, 2009. [74](#), [173](#)
- [159] P.R. Mouton. *Principles and practices of unbiased stereology : an introduction for bioscientists*. Johns Hopkins University Pr, 2002. [33](#)
- [160] A. Mraz, M. Gunaratne, A. Nazef, and B. Choubane. Experimental evaluation of a pavement imaging system : Florida department of transportation’s multipurpose survey vehicle. *Transportation Research Record : Journal of the Transportation Research Board*, 1974(1) :97–106, 2006. [166](#)
- [161] M. Nagao and T. Matsuyama. Edge preserving smoothing. *Computer Graphics and Image Processing*, 9(4) :394–407, 1979. [74](#)
- [162] M. Nagao, T. Matsuyama, and Y. Ikeda. Region extraction and shape analysis in aerial photographs. *Computer Graphics and Image Processing*, 10(3) :195–223, 1979. [95](#)
- [163] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *Image Processing, IEEE Transactions on*, 15(11) :3531–3539, 2006. [20](#), [116](#)
- [164] S. Nakagawa and I.C. Cuthill. Effect size, confidence interval and statistical significance : a practical guide for biologists. *Biological Reviews*, 82(4) :591–605, 2007. [158](#)
- [165] P. Nehlig. Applications quasi affines : pavages par images r ciproques. *Theoretical computer science*, 156(1-2) :1–38, 1996. [29](#)
- [166] J. Neyman and E.S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231 :289–337, 1933. [155](#)
- [167] N. Ngan, E. Dokl dalov , M. Akil, and F. Contou-Carr re. Fast and efficient FPGA implementation of connected operators. *Journal of Systems Architecture*, 57(8) :778–789, 2011. [118](#)
- [168] T.P. Nguyen, E. Parra-Denis, and M. Bilodeau. New multiresolution descriptor based on local binary pattern. Technical report, MINES ParisTech, Math matique et Syst me, CMM, 2012. [170](#), [171](#)

- [169] T.S. Nguyen, S. Begot, F. Duculty, and M. Avila. Free-form anisotropy : A new method for crack detection on pavement surface images. In *Image Processing, 18th International Conference on*, pages 1069–1072. IEEE, 2011. [12](#)
- [170] S. Nilsson. *Radix Sorting and Searching*. Thèse de Doctorat, Department of Computer Science, Lund University, Lund, Sweden, 1996. [119](#)
- [171] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1) :51–59, 1996. [170](#)
- [172] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7) :971–987, 2002. [170](#)
- [173] S. Outal, D. Jeulin, and J. Schleifer. A new method for estimating the 3D size distribution curve of fragmented rocks out of 2D images. *Image Analysis and Stereology*, 27 :97–105, 2008. [20](#)
- [174] G.K. Ouzounis and M.H.F. Wilkinson. Mask-based second-generation connectivity and attribute filters. *Pattern Analysis and Machine Intelligence, IEEE transactions on*, pages 990–1004, 2007. [123](#), [186](#)
- [175] G.K. Ouzounis and M.H.F. Wilkinson. Hyperconnected attribute filters based on k-flat zones. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2) :224–239, 2011. [100](#)
- [176] J. Paoli, M. Smedh, and M.B. Ericson. Multiphoton laser scanning microscopy—a novel diagnostic method for superficial skin cancers. In *Seminars in cutaneous medicine and surgery*, volume 28, pages 190–195. Elsevier, 2009. [145](#)
- [177] E. Parra-Denis. *Analyse morphologique 3D de particules de forme complexes : application aux intermétalliques dans les alliages d'aluminium*. Thèse de Doctorat, University Jean Monnet, Saint-Etienne, 2007. [103](#), [104](#)
- [178] D.V. Patel and C.N.J. McGhee. Contemporary in vivo confocal microscopy of the living human cornea using white light and laser scanning techniques : a major review. *Clinical & experimental ophthalmology*, 35(1) :71–88, 2007. [67](#)
- [179] E.S. Pearson, G.A. Barnard, and W.S. Gosset. 'Student'. Wiley Online Library, 1942. [156](#), [157](#)
- [180] J. Pecht. Speeding-up successive minkowski operations with bit-plane computers. *Pattern Recognition Letters*, 3(2) :113–117, 1985. [19](#)
- [181] A.M. Pena, M. Strupler, T. Boulesteix, and M. Schanne-Klein. Spectroscopic analysis of keratin endogenous signal for skin multiphoton microscopy. *Optics express*, 13(16) :6268–6274, 2005. [145](#)
- [182] K. Perlin. course in "advanced image synthesis". In *Special Interest Group on Computer Graphics and Interactive Techniques, Association for the Computing Machinery*, 1984. [107](#)
- [183] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7) :629–639, 1990. [74](#)
- [184] C. Peyrega. *Prédiction des propriétés acoustiques de matériaux fibreux hétérogènes à partir de leur microstructure 3D*. Thèse de Doctorat, École des Mines de Paris (MINES-ParisTech), 2010. [74](#)
- [185] P. Radchenko and G.M. James. Improved variable selection with forward-lasso adaptive shrinkage. *The Annals of Applied Statistics*, 5(1) :427–448, 2011. [129](#)
- [186] J. Radon. Über die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten. *Classic papers in modern diagnostic radiology*, page 5, 2005. [14](#)
- [187] J. Reunanen. Overfitting in making comparisons between variable selection methods. *The Journal of Machine Learning Research*, 3 :1371–1382, 2003. [129](#)
- [188] K. Robinson and P.F. Whelan. Efficient morphological reconstruction : a downhill filter. *Pattern Recognition Letters*, 25(15) :1759–1767, 2004. [56](#)
- [189] F. Roli. Measure of texture anisotropy for crack detection on textured surfaces. *Electronics Letters*, 32(14) :1274–1275, 1996. [12](#)

- [190] C. Ronse. Blood vessel detection through ridge skeletonization. Technical report, Report R482, Philips Research Laboratory, Brussels, 1984. [14](#)
- [191] C. Ronse. Erosion of narrow image features by combination of local low rank and max filters. In *Image Processing and its Applications, Proceedings 2nd International Conference on*, pages 77–81, 1986. [9](#)
- [192] C. Ronse and H.J.A.M. Heijmans. The algebraic basis of mathematical morphology : II. openings and closings. *Computer Vision, Graphics, and Image Processing*, 54(1) :74–97, 1991. [9](#)
- [193] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, 35(3) :1012–1030, 2007. [128](#)
- [194] P. Salembier. Structuring element adaptation for morphological filters. *Journal of Visual Communication and Image Representation*, 3(2) :115–136, 1992. [74](#)
- [195] P. Salembier. Connected operators based on region-trees. In *Image Processing, 15th International Conference on*, pages 2176–2179. IEEE, 2008. [99](#)
- [196] P. Salembier. Study on nonlocal morphological operators. In *Image Processing, 16th International Conference on*, pages 2269–2272. IEEE, 2009. [74](#)
- [197] P. Salembier and L. Garrido. Connected operators based on region-tree pruning. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 169–178. Springer, 2000. [20](#)
- [198] P. Salembier, A. Oliveras, and L. Garrido. Motion connected operators for image sequences. In *Signal Processing VIII Theories and applications*, volume 2, pages 1083–1086, 1996. [100](#)
- [199] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *Image Processing, IEEE Transactions on*, 7 :555–570, 1998. [20](#), [100](#), [116](#)
- [200] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *Image Processing, IEEE Transactions on*, 4(8) :1153–1160, 1995. [20](#), [98](#)
- [201] M. Schmitt. *Des algorithmes morphologiques à l'intelligence artificielle*. Thèse de Doctorat, Ecole Nationale Supérieure des Mines de Paris, 1989. [104](#)
- [202] J. Serra. *Image analysis and mathematical morphology*, volume 1. Academic, London, 1982. [9](#), [19](#), [21](#), [36](#)
- [203] J. Serra. *Image analysis and mathematical morphology*, volume 2 Theoretical Advances. Academic, London, 1988. [19](#), [74](#), [84](#), [98](#)
- [204] J. Serra and L. Vincent. An overview of morphological filtering. *Circuits System Signal Processing*, 11(1) :47–108, 1992. [98](#)
- [205] H. Shikata, E.A. Hoffman, and M. Sonka. Automated segmentation of pulmonary vascular tree from 3D CT images. In *Proceedings of SPIE*, volume 5369, page 107, 2004. [12](#)
- [206] P. Soille. Generalized geodesic distances applied to interpolation and shape description. *Mathematical Morphology and its Applications to Image Processing*, pages 193–200, 1994. [101](#), [102](#)
- [207] P. Soille. Generalized geodesy via geodesic time. *Pattern Recognition Letters*, 15(12) :1235–1240, 1994. [102](#)
- [208] P. Soille. Morphological partitioning of multispectral images. *Journal of Electronic Imaging*, 5(03) :252–265, 1996. [95](#)
- [209] P. Soille. On morphological operators based on rank filters. *Pattern recognition*, 35(2) :527–535, 2002. [9](#)
- [210] P. Soille. *Morphological image analysis : principles and applications*. Springer-Verlag New York, Inc., 2003. [19](#)
- [211] P. Soille. Morphological carving. *Pattern Recognition Letters*, 25(5) :543–550, apr 2004. [102](#)
- [212] P. Soille. Constrained connectivity for hierarchical image partitioning and simplification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7) :1132–1145, 2008. [95](#)
- [213] P. Soille, E.J. Breen, and R. Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(5) :562–567, 1996. [29](#), [34](#), [41](#)

- [214] P. Soille and J. Grazzini. Advances in constrained connectivity. In *Discrete Geometry for Computer Imagery*, pages 423–433. Springer, 2008. [95](#)
- [215] P. Soille and H. Talbot. Directional morphological filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11) :1313–1329, 2001. [9](#), [31](#)
- [216] I. Sokolic. *Criteria to evaluate the quality of pavement camera systems in automated evaluation vehicles*. Thèse de Doctorat, University of South Florida, 2004. [166](#)
- [217] J. Stawiaski and E. Decenci re. Region merging via graph-cuts. *Image Analysis and Stereology*, 27(1) :39–45, 2008. [146](#)
- [218] C. Steger. An unbiased detector of curvilinear structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(2) :113–125, 1998. [12](#)
- [219] S.R. Sternberg. Grayscale morphology. *Computer Vision, Graphics, and Image Processing*, 35(3) :333–355, 1986. [84](#)
- [220] G.C. Stockman and A.K. Agrawala. Equivalence of Hough curve detection to template matching. *Communications of the Association for the Computing Machinery*, 20(11) :820–822, 1977. [14](#)
- [221] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974. [136](#)
- [222] C. Sun and P. Vallotton. Fast linear feature detection using multiple directional non-maximum suppression. *Journal of Microscopy*, 234(2) :147–157, 2009. [9](#)
- [223] H. Talbot. A morphological algorithm for linear segment detection. *Mathematical Morphology and its Applications to Signal Processing*, pages 219–226, 1996. [14](#)
- [224] H. Talbot and B. Appleton. Efficient complete and incomplete path openings and closings. *Image and Vision Computing*, 25(4) :416–425, 2007. [44](#), [52](#), [122](#)
- [225] O. Tankyevych. *Filtering of thin objects : applications to vascular image analysis*. Th se de Doctorat, University Paris-Est, ESIEE, 2010. [8](#)
- [226] O. Tankyevych, H. Talbot, and P. Dokl dal. Curvilinear morpho-hessian filter. In *Biomedical Imaging : From Nano to Macro. 5th IEEE International Symposium on*, pages 1011–1014. IEEE, 2008. [14](#), [44](#)
- [227] O. Tankyevych, H. Talbot, P. Dokl dal, and N. Passat. Spatially-Variant Morpho-Hessian filter : Efficient implementation and application. In *Mathematical Morphology and Its Application to Signal and Image Processing*, pages 137–148. Springer, 2009. [74](#)
- [228] R.E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the Association for the Computing Machinery*, 22(2) :215–225, 1975. [118](#)
- [229] N. Theera-Umpon and P.D. Gader. Counting white blood cells using morphological granulometries. *Journal of Electronic Imaging*, 9 :170, 2000. [20](#)
- [230] G. Thibault. *Indices de formes et de texture : de la 2D vers la 3D. Application au classement de noyaux de cellules*. Th se de Doctorat, Facult  des sciences de Luminy, Universit  de la M diterran e, Aix-Marseille II, 2009. [172](#)
- [231] G. Thibault, B. Fertil, C. Navarro, S. Pereira, P. Cau, N. Levy, J. Sequeira, and J.L. Mari. Texture indexes and gray level size zone matrix. application to cell nuclei classification. *Pattern Recognition and Information Processing*, pages 140–145, 2009. [172](#)
- [232] G. Thibault, X. Zhang, P. Dokl dal, Marcotegui B., G. Cazuguel, B. Lay, P. Massin, and E. Decenci re. Multi-scale gabor filter for blood vessel detection in clinical databases. In *Pattern Recognition, International Conference on*, IEEE, 2012. [11](#)
- [233] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. [128](#)
- [234] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision. Sixth International Conference on*, pages 839–846. IEEE, 2002. [9](#), [74](#)



## BIBLIOGRAPHIE

---

- [235] P.T.H. Truc, M.A.U. Khan, Y.K. Lee, S. Lee, and T.S. Kim. Vessel enhancement filter using directional filter bank. *Computer Vision and Image Understanding*, 113(1) :101–112, 2009. [12](#)
- [236] J.W. Tukey. Exploratory data analysis. *Reading, MA*, 1977. [9](#)
- [237] E.R. Urbach and M.H.F. Wilkinson. Shape-only granulometries and gray-scale shape filters. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, page 305, 2002. [98](#), [100](#)
- [238] S. Valero, J. Chanussot, JA Benediktsson, H. Talbot, and B. Waske. Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Letters*, 31(10) :1120–1127, 2010. [44](#)
- [239] M. Van Droogenbroeck and M.J. Buckley. Morphological erosions and openings : fast algorithms based on anchors. *Journal of Mathematical Imaging and Vision*, 22(2) :121–142, 2005. [18](#), [19](#), [32](#), [36](#)
- [240] M. Van Droogenbroeck and H. Talbot. Fast computation of morphological operations with arbitrary structuring elements. *Pattern Recognition Letters*, 17(14) :1451–1460, 1996. [94](#)
- [241] M. Van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13(7) :517–521, 1992. [19](#), [36](#)
- [242] R. Verdú-Monedero, J. Angulo, and J. Serra. Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields. *Image Processing, IEEE Transactions on*, 20(1) :200–212, 2011. [74](#)
- [243] L. Vincent. Grayscale area openings and closings, their efficient implementation and applications. In *Mathematical Morphology and its Applications to Signal Processing, First Workshop on*, pages 22–27. Springer, 1993. [78](#), [98](#)
- [244] L. Vincent. Morphological grayscale reconstruction in image analysis : Applications and efficient algorithms. *Image Processing, IEEE Transactions on*, 2(2) :176–201, 1993. [49](#), [56](#), [98](#)
- [245] L. Vincent. Minimal path algorithms for the robust detection of linear features in gray images. *Computational Imaging and Vision*, 12 :331–338, 1998. [12](#)
- [246] L. Vincent. Granulometries and opening trees. *Mathematical morphology*, page 57, 2000. [19](#), [39](#)
- [247] L. Vincent and P. Soille. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE transactions on*, pages 583–598, 1991. [14](#), [48](#)
- [248] J.H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, pages 236–244, 1963. [174](#)
- [249] J. Weber, S. Lefèvre, and P. Gañçarski. Spatio-temporal quasi-flat zones for morphological video segmentation. *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 178–189, 2011. [95](#)
- [250] B.L. Welch. The generalization of Student’s’ problem when several different population variances are involved. *Biometrika*, 34(1/2) :28–35, 1947. [157](#)
- [251] M.B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1) :1–17, 1968. [155](#)
- [252] M.H.F. Wilkinson, H. Gao, W.H. Hesselink, J.E. Jonker, and A. Meijster. Concurrent computation of attribute filters on shared memory parallel machines. *Pattern Analysis and Machine Intelligence, IEEE transactions on*, pages 1800–1813, 2008. [42](#)
- [253] M.H.F. Wilkinson and J.B.T.M. Roerdink. Fast morphological attribute operations using Tarjan’s union-find algorithm. *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 311–320, 2002. [118](#)
- [254] M.H.F. Wilkinson and M.A. Westenberg. Shape preserving filament enhancement filtering. *Medical Image Computing and Computer-assisted Intervention*, 2208 :770–777, 2001. [98](#)
- [255] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67, 2006. [129](#)

- [256] W.R. Zipfel, R.M. Williams, R. Christie, A.Y. Nikitin, B.T. Hyman, and W.W. Webb. Live tissue intrinsic emission microscopy using multiphoton-excited native fluorescence and second harmonic generation. *Proceedings of the national academy of sciences of the United States of America*, 100(12) :7075, 2003. [145](#)
- [257] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476) :1418–1429, 2006. [129](#), [135](#)
- [258] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005. [129](#)

## Détection de structures fines par traitement d'images et apprentissage statistique : application au contrôle non destructif

**Résumé :** Dans cette thèse, nous présentons de nouvelles méthodes de traitement d'images pour extraire ou rehausser les éléments fins d'une image. Pour ces opérateurs, issus de la morphologie mathématique, l'accent a été mis principalement sur la précision de détection et sur le temps de calcul, qui doivent être optimisés pour pouvoir répondre aux contraintes de temps imposées par différentes applications industrielles. La première partie de ce mémoire présente ces méthodes, organisées en fonction de la tortuosité des objets à détecter. Nous commençons par proposer un algorithme rapide pour le calcul des ouvertures 1-D afin d'extraire des structures rectilignes des images. Puis, nous étudions une nouvelle classe d'opérateurs rapides avec les ouvertures parcimonieuses par chemins, permettant d'analyser des structures ayant une tortuosité modérée. Enfin, nous proposons de nouveaux éléments structurants adaptatifs et des filtres connexes construits avec des attributs géodésiques et géométriques pour extraire des structures filiformes ayant une tortuosité quelconque.

Dans un second temps, nous avons développé une méthode d'analyse statistique en introduisant une nouvelle pénalisation adaptative. L'objectif consiste à créer un modèle prédictif précis, qui minimise en même temps une fonction de coût, indépendante des données. Lorsque cette fonction de coût est liée au temps de calcul de chaque descripteur, il est alors possible de créer un modèle parcimonieux précis et qui minimise les temps de calcul. Cette méthode est une généralisation des régressions linéaires et logistiques Ridge, Forward stagewise, Lar, ou Lasso.

Les algorithmes développés dans cette thèse ont été utilisés pour trois applications industrielles, très différentes les unes des autres, mais toutes faisant intervenir une approche multidisciplinaire : le traitement d'images et l'analyse statistique. L'association de ces deux disciplines permet d'améliorer la généralité des stratégies proposées puisque les opérateurs de traitement d'images alliés à un apprentissage supervisé ou non supervisé, permettent d'adapter le traitement à chaque application.

**Mots clés :** Traitement d'images, morphologie mathématique, analyse statistique, caractérisation de formes, contrôles non destructifs, ouvertures parcimonieuses par chemins, region growing structuring elements, amincissements par attributs géodésiques et topologiques, adaptive coefficient shrinkage.

## Thin structures detection by means of image processing and statistical learning: application to non-destructive testing

**Abstract:** This PhD is dedicated to new image processing methods to extract or enhance thin objects from an image. These methods stem from mathematical morphology, and they mainly focus on the accuracy of the detection and on the computation time. This second constraint is imposed by the fact that we are dealing with high-throughput applications. The first part of this thesis presents these methods, organized according to the tortuosity of the objects to detect. We first propose a fast algorithm for the computation of 1-D openings, used to extract thin and straight structures in the images. Then, we study a new class of fast operators, parsimonious path openings, which can extract thin structures with moderate tortuosities. Finally, we propose new adaptive structuring elements and new thinnings with geodesic and geometric attributes to filter out the noise and to enhance thin structures of any tortuosity.

Besides, we have developed a machine learning method by introducing a new adaptive penalization. We aim at creating a predictive model that minimizes a cost function (independent of the data) while preserving a good accuracy. When this cost function is linked to the computation time of each feature, the resulting models will optimize the timings, while preserving a good accuracy. This method is a generalization of linear and logistic regressions with Ridge, Forward stagewise, Lar or Lasso penalization.

The algorithms developed in this thesis have been used for three industrial applications. While their objectives are very different, the framework is the same (non-destructive testing) and they all involve a multidisciplinary approach (images processing and statistical analysis). The combination of these two fields yields a higher flexibility in comparison with classical methods. Generic strategies are used, since image processing operators are associated to statistical learning (supervised or unsupervised) to make a specific treatment for each application.

**Keywords:** Image processing, mathematical morphology, statistical analysis, pattern recognition, non destructive testing, parsimonious path openings, region growing structuring elements, geodesic and topologic attributes thinnings, adaptive coefficient shrinkage.

